Valuation of Multiple Exercise Options with Energy Applications

Selvaprabu Nadarajah, François Margot, Nicola Secomandi
Tepper School of Business, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213-3890, USA
{snadaraj, fmargot, ns7}@andrew.cmu.edu
Tepper Working Paper 2012-E54
September 2012

Abstract

We develop least squares Monte Carlo (LSM) and approximate linear programming (ALP) methods for valuing multiple exercise options, such as energy swing and storage options, using term structure models. Our numerical and theoretical investigation shows the superiority of a rarely used LSM variant for estimating lower and upper bounds on the option value over the standard LSM version and the ALP approach. We also structurally relate the seemingly different LSM and ALP methods using the concept of surrogate relaxations. This analysis motivates further research into surrogate relaxations of approximate linear programs.

1 Introduction

The pricing of options with multiple exercise is an important area of financial engineering, with applications including commodity, energy, and interest rate derivatives. Examples include chooser flexible caps (Meinshausen and Hambly 2004), portfolio liquidation (Gyurko et al. 2011), swing options (Barbieri and Garman 1996, Jaillet et al. 2004, Chandramouli and Haugh 2012), switching options (Cortazar et al. 2008), and commodity processing and storage (Maragos 2002, Boogert and De Jong 2008, Secomandi 2010, Lai et al. 2010, Boogert and Mazières 2011, Devalkar et al. 2011, Nadarajah et al. 2012, Thompson 2012, Wu et al. 2012). In particular, our focus is on energy swing and storage options.

Term structure models are widespread among practitioners in commodity, energy, and fixed income industries (Ho and Lee 1986, Cortazar and Schwartz 1994, Clewlow and Strickland 2000, Maragos 2002, Eydeland and Wolyniec 2003, Veronesi 2010). Valuing multiple exercise options using such models generally gives rise to an intractable Markov decision problem (MDP), which we refer to as the valuation MDP, due to two curses of dimensionality: (i) The high dimensionality of the MDP state space and (ii) the computation of high dimensional expectations (Powell 2011, §4.1).

The financial engineering literature typically approaches the solution of these MDPs using Monte Carlo based approximate dynamic programming (ADP) techniques (see Rogers 2002, Chapter 8 in Glasserman 2004, Detemple 2006, Andersen and Broadie 2004, Brown et al. 2010, and references therein). These methods compute a heuristic exercise policy and lower and upper bounds on the option value.

The least squares Monte Carlo (LSM) approach, pioneered by Carriere (1996), Longstaff and Schwartz (2001), and Tsitsiklis and Van Roy (2001), has become the norm for valuing multiple exercise options (see Appendix B in Eydeland and Wolyniec 2003, Glasserman and Yu 2004, Meinshausen and Hambly 2004, Bender 2011, Gyurko et al. 2011). This method approximates the valuation MDP continuation (value) function, using a convenient approximation to overcome the

second curse of dimensionality (Tsitsiklis and Van Roy 2001). We thus refer to this method as LSMC (where C stands for continuation).

A largely ignored version of the LSM approach approximates the value function of the valuation MDP, rather than its continuation value function. We refer to this LSM variant as LSMV (where V abbreviates value). LSMV was proposed by Bellman and Dreyfus (1959) more than half a century ago, but not in the context of option pricing. We are not aware of the use of LSMV for pricing multiple exercise options. It is thus unclear how LSMV compares to LSMC for valuing these options.

Another ADP approach is approximate linear programming (ALP; Schweitzer and Seidmann 1985, de Farias and Van Roy 2003). ALP approximates the value function of an MDP. ALP has received some attention in operations research (Adelman 2007, Farias and Van Roy 2007, Moallemi et al. 2008, Tong and Topaloglu 2011, Vossen and Zhang 2012), but its application to option valuation is scant (Nadarajah et al. 2012). We are not aware of any study that compares the LSM and ALP approaches for valuing multiple exercise options.

The goal of this paper is twofold: (1) Develop LSMV and ALP methods to price multiple exercise options when using term structure models and (2) benchmark the performance of these methods against the one of LSMC on two applications: Energy swing and storage options.

A difficulty with LSMV and ALP is the computation of high dimensional expectations in the valuation MDP. We address this issue by approximating value functions using basis functions for which these expectations can be computed in essentially *closed form*. Basis functions such as polynomials of futures prices and prices of call and put options on futures prices belong to this class. The LSMV and ALP methods that we develop thus avoid the approximation that LSMC makes to overcome this difficulty.

ALP requires the solution of a large scale linear program. This can be challenging. For example, Adelman (2007) uses column generation to solve his linear programs. In contrast, we develop an efficient decomposition technique to heuristically solve our linear programs. We label the resulting ALP method as ALPD (where D stands for decomposition).

We benchmark ALPD and LSMV against LSMC on realistic energy swing and storage option instances, computing lower and upper bounds on the value of these real options using each method.

We observe that LSMV estimates tighter lower bounds than ALPD and obtains them in a faster fashion. The tightest and near optimal LSMV lower bound estimates are comparable to the tightest ones from LSMC. However, LSMV requires an order of magnitude fewer regression samples than LSMC to obtain these estimates. As a result, the overall LSMV CPU time to estimate its tightest lower bounds is on average smaller than the analogous LSMC requirement by 1.32 and 2.28 on the swing and storage option instances, respectively.

The upper bounds estimated by LSMV are tighter than the ones estimated by ALPD but comparable to the ones obtained by LSMC. Estimating upper bounds using LSMV and ALPD is more than two orders of magnitude faster than estimating these bounds using LSMC. Glasserman and Yu (2004) and Gyurko et al. (2011) propose LSMC variants that are likely faster than LSMC for estimating upper bounds, but do not quantify the possible computational advantage. Further,

as LSMC, these methods involve approximations to overcome the second curse of dimensionality, whereas LSMV does not. Thus, we do not benchmark LSMV against these LSMC variants.

Overall, our numerical investigation provides strong support for using LSMV for the valuation of multiple exercise options as a method that is more *accurate* than ALPD, more *precise* than LSMC, and more *computationally efficient* than both ALPD and LSMC.

We also derive ∞-norm error bounds on the approximations determined by LSMV, LSMC, and a more analytically tractable version of ALPD. These bounds pinpoint the source of errors when using each method and provide some theoretical support for our numerical results. Furthermore, we identify a structural connection between LSMV and our ALP methods using the concept of surrogate relaxations (Glover 1968). Nadarajah et al. (2012) develop ALP surrogate relaxations for commodity storage valuation. Our analysis yields what appears to be the first structural connection between LSM and ALP methods.

The remainder of this paper is organized as follows. In §2 we introduce our general MDP formulation and its specification to energy swing and storage options. In §3 we discuss the bounding framework that we apply. In §4 we define value function approximations (VFAs) and continuation VFAs (CVFAs), and describe a family of VFAs that overcome the second curse of dimensionality. We develop LSMV and ALPD in §5 and §6, respectively, also describing LSMC in §5. We discuss the results of our numerical study in §7. We perform our error bound and surrogate relaxation analyses in §8 and §9, respectively. We conclude in §10. All proofs are in Appendix A.

2 Valuation MDP

We describe our valuation MDP in §2.1. We discuss two applications of this model in §2.2: The valuation of energy swing and storage options.

2.1 Model

We formulate the problem of valuing a multiple exercise option as an MDP. There is a finite number of times T_i, $i \in \mathcal{I} := \{0, \ldots, N-1\}$. The set $\mathcal{I}$ is the stage set. The state of our MDP at stage i is partitioned into *endogenous* and *exogenous* components. The endogenous component is the scalar x_i. It belongs to a discrete set $\mathcal{X}_i$ that represents information about the number of remaining exercise rights at stage i. The *exogenous* component is the vector $F_i \in \mathbb{R}^{N-i}$ that represents the option underlying term structure $(F_{i,i}, F_{i,i+1}, \ldots, F_{i,N-1})$, where $F_{i,j}$ is the element of the term structure associated with date T_j at time T_i. In commodity and energy applications, F_i is the forward curve, $F_{i,i}$ is the time T_i spot price, and $F_{i,j}$ is the time T_i futures price with maturity at time $T_j > T_i$. For applications involving interest rate derivatives, F_i is a bond yield curve and $F_{i,j}$ is the time T_i interest rate of the bond with maturity at date T_j.

At stage i and state (x_i, F_i), the decision maker chooses an exercise action a_i from the set $\mathcal{A}_i(x_i)$ and receives an immediate reward $r_i : (a_i, F_i) \mapsto \mathbb{R}$. The discrete set $\mathcal{A}_i(x_i)$ includes the number of rights that can be exercised at stage i. Subsequently, the endogenous part of the state transitions from x_i to $x_{i+1} := x_i - a_i$, and the exogenous part of the state evolves from F_i to $\tilde{F}_{i+1}$ according to a known risk-adjusted stochastic process (we use $\tilde{\cdot}$ to denote random quantities).

Let $\mathbb{E}$ denote expectation under a risk-adjusted probability measure for this process. A policy π is the collection of decision functions $\{A_0^\pi, \ldots, A_{N-1}^\pi\}$, where $A_i^\pi : (x_i, F_i) \mapsto \mathcal{A}_i(x_i)$, $\forall i \in \mathcal{I}$, $(x_i, F_i) \in \mathcal{X}_i \times \mathbb{R}^{N-i}$. We let Π be the set of all feasible policies. We denote by δ the risk-free discount factor from each time T_i back to time T_{i-1}, $i \in \mathcal{I} \setminus \{0\}$, that is, the discount factor is constant across stages. This assumption can be relaxed in a straightforward manner. Our stochastic optimization model is

$$\max_{\pi \in \Pi} \sum_{i \in \mathcal{I}} \delta^i \mathbb{E}\left[r_i(A_i^\pi(\tilde{x}_i^\pi, \tilde{F}_i), \tilde{F}_i) | x_0, F_0 \right],$$

where x_i^π is the random endogenous part of the state at stage i when using policy π.

This model can be equivalently reformulated as the following valuation MDP, $\forall i \in \mathcal{I}$ and $(x_i, F_i) \in \mathcal{X}_i \times \mathbb{R}^{N-i}$:

$$V_i(x_i, F_i) = \max_{a \in \mathcal{A}_i(x_i)} r_i(a, F_i) + \delta \mathbb{E}\left[V_{i+1}\left(x_i - a, \tilde{F}_{i+1} \right) | F_i' \right], \tag{1}$$

with boundary conditions $V_N(x_N, F_N) := 0$, $\forall x_N \in \mathcal{X}_N$, where $V_i(x_i, F_i)$ is the optimal value function in stage i and state (x_i, F_i), $F_i' := (F_{i,i+1}, \ldots, F_{i,N-1})$, and we assume that F_i' is sufficient to compute the expectation in (1) (this is true for common term structure models, including the ones used in this paper). We refer to MDP (1) as the *value function* MDP formulation. Given the initial state (x_0, F_0), the option value is $V_0(x_0, F_0)$.

The continuation value function $C_i(x_{i+1}, F_i')$, $\forall i \in \mathcal{I} \setminus \{N-1\}$ and $(x_{i+1}, F_i') \in \mathcal{X}_{i+1} \times \mathbb{R}^{N-i-1}$, is defined as

$$C_i(x_{i+1}, F_i') := \delta \mathbb{E}\left[V_{i+1}\left(x_{i+1}, \tilde{F}_{i+1} \right) | F_i' \right]. \tag{2}$$

By substituting the right hand side of (1) expressed for $i+1$ into (2) we obtain the following *continuation value function* MDP formulation, $\forall i \in \mathcal{I} \setminus \{N-1\}$ and $(x_{i+1}, F_i') \in \mathcal{X}_{i+1} \times \mathbb{R}^{N-i-1}$ as

$$C_i(x_{i+1}, F_i') = \delta \mathbb{E}\left[\max_{a \in \mathcal{A}_{i+1}(x_{i+1})} r_{i+1}(a, \tilde{F}_{i+1}) + C_{i+1}\left(x_{i+1} - a, \tilde{F}_{i+1}' \right) | F_i' \right], \tag{3}$$

with boundary conditions $C_{N-1}(x_N, F_{N-1}') := 0$, $\forall x_N \in \mathcal{X}_N$. In this case, the option value is computed as $V_0(x_0, F_0) = \max_{a \in \mathcal{A}_0(x_0)} r_0(x_0, F_0) + C_0(x_0 - a, F_0')$.

In the rest of this paper, we assume that the term structure evolution is governed by the *risk neutral* (risk adjusted) dynamics of a continuous time multifactor model (Ho and Lee 1986, Cortazar and Schwartz 1994) that we now describe. Pick time $t \in [T_0 := 0, T_i)$ with $i \in \mathcal{I} \setminus \{0\}$. We denote by $F(t, T_m)$ the time t value of the m-th element of the term structure, that is, the one associated with date T_m. We define $F'(t) := (F(t, T_m), m = i, \ldots, N-1)$. Given a predetermined number $K \in \{1, \ldots, N-1\}$ of stochastic factors, the evolution of $F'(t)$ is governed by the following stochastic

differential equations:

$$\frac{dF(t,T_m)}{F(t,T_m)} = \sum_{k=1}^{K} \sigma_{m,k}(t) dW_k(t), \ \forall m \in \mathcal{I} \setminus \{0\}, \tag{4}$$

$$dW_k(t) dW_{k'}(t) = 0, \ \forall k, k' \in \{1, \ldots, K\}, \ k \neq k', \tag{5}$$

where $\sigma_{m,k}(t)$ is the time t loading coefficient on the Brownian motion W_k for the term structure element $F(t, T_m)$.

The term structure model (4)-(5) is quite general. It captures seasonality in the variances and covariances of changes in the term structure elements because the loading factors are time dependent, and seasonality in the term structure levels from the seasonality in the initial (time T_0) term structure. Low dimensional price models that are typically used with LSM methods for valuing energy and commodity multiple exercise real options (Meinshausen and Hambly 2004, Boogert and De Jong 2008, Bender 2011, Gyurko et al. 2011) are special cases of (4)-(5). Examples include the one factor model of Schwartz (1997) and the two factor model of Schwartz and Smith (2000).

2.2 Energy Applications

In this subsection, we illustrate the application of the valuation MDP (1) to energy swing and storage options, which are the focus of our numerical study in §7. The term structure in these applications is an energy forward curve. Thus, $F(t, T_m)$ is the time t price of the futures contract with maturity on date T_m.

Energy Swing Option. Swing options are commonly encountered in energy applications (Barbieri and Garman 1996, Jaillet et al. 2004). We focus on a purchase swing option. This option is used, for example, by a producer of ethylene that requires the amount $q_i > 0$ of crude oil as input to a thermal cracking process at each time T_i, $i \in \mathcal{I}$. The contract has two parts: A *purchase* part that involves buying the quantity q_i at the strike price K_i, $\forall i \in \mathcal{I}$; and a *swing* part that endows the producer with $n \leq N$ *swing rights* to increase or decrease each purchase amount q_i by a fixed constant $Q_i \in (0, q_i]$ at the strike price K_i at each stage i. At most one swing right can be exercised at a given stage i.

The incentive to exercise this swing option in stage i stems from the producer's ability to transact in the spot market at the prevailing spot price $F_{i,i}$. We assume that the energy procured at stage i is delivered over the interval $[T_i, T_{i+1})$. Therefore, if $K_i > F_{i,i}$, the producer has the incentive to purchase a quantity $q_i - Q_i$ from the purchase swing contract at the strike price K_i and purchase a quantity Q_i from the spot market at the price $F_{i,i}$. This combined trade results in an effective gain of $Q_i(K_i - F_{i,i})$ relative to procuring q_i at the strike price K_i. Similarly, if $K_i < F_{i,i}$, the producer has the incentive to purchase a quantity $q_i + Q_i$ from the purchase swing contract at the strike price K_i and sell a quantity Q_i into the spot market at price $F_{i,i}$, for a gain of $Q_i(F_{i,i} - K_i)$.

The swing part of this purchase swing option can be valued in our MDP framework by defining the endogenous state variable x_i to be the number of available swing rights at stage i. The set $\mathcal{X}_i$ is thus $\{i, \ldots, n\}$. The feasible action set is $\mathcal{A}_i(x_i) := \{0, 1\}$ if $x_i > 0$, and $\mathcal{A}_i(x_i) := \{0\}$ if

$x_i = 0$. That is, exercise is allowed only when there is at least one swing right available. The stage i immediate reward function $r_i(a, F_i)$ is defined as $Q_i \cdot |K_i - F_{i,i}| \cdot a$. It is the effective gain from exercising a swing right.

Energy Storage Option. Consider an energy (e.g., natural gas) storage facility with finite space $\bar{x}$ and finite injection and withdrawal capacities $\overline{a}$ and $\underline{a}$, which satisfy $0 > \overline{a} \geq -\bar{x}$ and $\bar{x} \geq \underline{a} > 0$ (see Secomandi 2010 and Lai et al. 2010 for details). The storage manager has the option to buy energy from the wholesale spot market and inject it into this facility or withdraw from this facility previously purchased and injected energy and sell it into the wholesale spot market.

The endogenous state x_i is the inventory in storage at stage $i \in \mathcal{I}$. It belongs to the interval $\mathcal{X}_i := [0, \bar{x}]$. At stage $i \in \mathcal{I}$, a positive action is an energy withdrawal and sell decision, a negative action is an energy purchase and inject decision, and zero is the do nothing decision. The set of feasible injections, withdrawals, and overall actions are $\mathcal{A}_i^I(x_i) := [\max\{\underline{a}, (x_i - \bar{x})\}, 0]$, $\mathcal{A}_i^W(x_i) := [0, \min\{x_i, \overline{a}\}]$, and $\mathcal{A}_i(x_i) := \mathcal{A}_i^I(x_i) \cup \mathcal{A}_i^W(x_i)$, respectively. Although the sets $\mathcal{X}_i$, $\mathcal{A}_i^I(x_i)$, and $\mathcal{A}_i^W(x_i)$ are intervals, by Lemma 1 in Secomandi et al. (2012) they can be optimally discretized if $\underline{a}$, $\overline{a}$, and $\bar{x}$ are rational. We assume this to be the case in this paper.

Let the coefficients $\alpha^W \in (0, 1]$ and $\alpha^I \geq 1$ model energy losses associated with energy withdrawals and injections, respectively, and the coefficients ς^W and ς^I represent withdrawal and injection marginal costs, respectively. The immediate reward function is $r_i(a_i, F_i) := (\alpha^I F_{i,i} + \varsigma^I)a_i$ if $a_i \in \mathbb{R}_-$; $r_i(a_i, F_i) := 0$ if $a_i = 0$; and $r_i(a_i, F_i) := (\alpha^W F_{i,i} - \varsigma^W)a_i$ if $a_i \in \mathbb{R}_+$.

3 Bounding the Option Value

Computing the exact option value $V_0(x_0, F_0)$ by solving MDPs (1) or (3) is typically intractable, as discussed in §1. Therefore, in this section we discuss an approach for estimating bounds on the option value using the *given* low dimensional VFA $\hat{V}_i(x_i, F_i)$ and CVFA $\hat{C}_i(x_{i+1}, F_i')$. These functions allow one to heuristically break the first curse of dimensionality, as now discussed.

To estimate a lower bound on the option value $V_0(x_0, F_0)$ we generate a set of W term structure sample paths $\{F_i^w, i \in \mathcal{I}, w = 1, \ldots, W\}$ starting from the term structure F_0 at time T_0, and simulate the greedy policy induced by the VFA or CVFA. That is, on each sample path and at each stage i and state (x_i, F_i), a greedy action is computed by solving

$$\max_{a \in \mathcal{A}_i(x_i)} r_i(a, F_i) + \delta \mathbb{E}[\hat{V}_{i+1}(x_i - a, \tilde{F}_{i+1}) | F_i']$$

when using a VFA and

$$\max_{a \in \mathcal{A}_i(x_i)} r_i(a, F_i) + \hat{C}_i(x_i - a, F_i')$$

when using a CVFA. We estimate a lower bound on the option value by averaging the time T_0 discounted values of all the rewards obtained from implementing the greedy actions computed along each sample path.

Evaluating $\mathbb{E}[\hat{V}_{i+1}(x_i - a, \tilde{F}_{i+1}) | F_i']$ typically poses a challenge (this is the second curse of dimensionality). We discuss in §4 classes of VFAs that overcome this issue, thus allowing us to efficiently

estimate VFA based lower bounds.

To evaluate the quality of the greedy lower bounds, we estimate upper bounds by Monte Carlo simulation using the information relaxation and duality framework (see Brown et al. 2010, and references therein). We use the same set of W term structure sample paths $\{F_i^w, i \in \mathcal{I}, w = 1, \ldots, W\}$ already generated for lower bound estimation. This approach relies on the availability of dual feasible penalties $p_i(x_{i+1}, F_{i+1}, F_i')$ that penalize knowledge at time T_i of the future information F_{i+1}: The feasibility requirement is $\mathbb{E}[p_i(x_{i+1}, \tilde{F}_{i+1}, F_i')|F_i'] \leq 0$ (see Brown et al. 2010 for details). Such penalties can be defined using a VFA as follows:

$$\hat{V}_{i+1}(x_{i+1}, F_{i+1}) - \mathbb{E}\left[\hat{V}_{i+1}(x_{i+1}, \tilde{F}_{i+1})|F_i'\right]. \tag{6}$$

When using a CVFA, the natural analogue to (6) is

$$\left[\max_{a \in \mathcal{A}_{i+1}(x_{i+1})} r_{i+1}(a, F_{i+1}) + \hat{C}_{i+1}(x_{i+1} - a, F_{i+1}')\right] - \hat{C}_i(x_{i+1}, F_i')/\delta. \tag{7}$$

When using the penalties (7), it holds that

$$\begin{aligned}\mathbb{E}\left[\max_{a \in \mathcal{A}_{i+1}(x_{i+1})} \left\{r_{i+1}(a, \tilde{F}_{i+1}) + \hat{C}_{i+1}(x_{i+1} - a, \tilde{F}_{i+1}')\right\}|F_i'\right] - \hat{C}_i(x_{i+1}, F_i')/\delta \\ \geq \mathbb{E}[\hat{C}_{i+1}(x_{i+1}, \tilde{F}_{i+1}')|F_i'] - \hat{C}_i(x_{i+1}, F_i')/\delta,\end{aligned} \tag{8}$$

where the last inequality is obtained by setting $a = 0$ and assuming $r_{i+1}(0, F_{i+1}) = 0$, $\forall F_{i+1} \in \mathbb{R}^{N-i-1}$. This assumption is true for the applications discussed in §2.2. In general, the right hand side of (8) can be strictly positive. This implies that the dual penalties (7) are not guaranteed to be feasible and can lead to invalid upper bounds on the option value. Thus, it is not advisable to use them.

In contrast, dual feasible penalties can be defined by using the VFA induced by a CVFA as follows:

$$\begin{aligned}\max_{a \in \mathcal{A}_{i+1}(x_{i+1})} \left\{r_{i+1}(a, F_{i+1}) + \hat{C}_{i+1}\left(x_{i+1} - a, F_{i+1}'\right)\right\} \\ -\mathbb{E}\left[\max_{a \in \mathcal{A}_{i+1}(x_{i+1})} \left\{r_{i+1}(a, \tilde{F}_{i+1}) + \hat{C}_{i+1}\left(x_{i+1} - a, \tilde{F}_{i+1}'\right)\right\}|F_i'\right].\end{aligned} \tag{9}$$

However, computing the penalties (9) is computationally burdensome, especially because the second term of (9) contains a maximization inside an expectation. Thus, a comparison of (6) and (9) indicates that VFAs have a possibly substantial computational advantage over CVFAs when computing dual feasible penalties, and hence estimating upper bounds.

Once dual feasible penalties are specified, the point estimate $U_0^w(x_0)$ of an upper bound on the option value $V_0(x_0, F_0)$ can be obtained by solving the following dynamic program defined on the

w-th term structure sample path:

$$U_i^w(x_i) = \max_{a \in \mathcal{A}_i(x_i)} r_i(a, F_i^w) - p_i(x_i - a, F_{i+1}^w, F_i'^w) + \delta U_{i+1}^w(x_i - a),$$

$\forall i \in \mathcal{I}$ and $x_i \in \mathcal{X}_i$, with boundary conditions $U_N^w(x_N) := 0$, $\forall x_N \in \mathcal{X}_N$. An upper bound estimate on the option value $V_0(x_0, F_0)$ is obtained by averaging the point estimates $U_0^w(x_0)$, $\forall w \in \{1, \ldots, W\}$.

4 VFAs and CVFAs

The bounding approach discussed in §3 requires the availability of a VFA or a CVFA. In this section, we define VFAs and CVFAs that are linear combinations of *basis functions* defined on subsets of the state space. These types of functions are commonly used in the ADP literature (e.g., see Chapter 8 in Bertsekas 2007 and page 326 in Powell 2011). We also discuss classes of basis functions that allow us to overcome the second curse of dimensionality when using a VFA.

Define the set of VFA basis functions as $\Phi_i := \{\phi_{i,b}, b = 1, \ldots, B_i^V\}$, $\forall i \in \mathcal{I} \setminus \{0\}$, where $\phi_{i,b} : (x_i, F_i) \mapsto \mathbb{R}$ is a basis function. A VFA is defined by a linear combination of the basis functions in Φ_i using the vector of coefficients $\beta_i := \{\beta_{i,1}, \ldots, \beta_{i,B_i^V}\}$ as

$$\hat{V}_i(x_i, F_i; \beta_i) := \sum_{b=1}^{B_i^V} \phi_{i,b}(x_i, F_i)\beta_{i,b} =: (\Phi_i \beta_i)(x_i, F_i).$$

Analogously, define the set of CVFA basis functions $\Psi_i = \{\psi_{i,b}, b = 1, \ldots, B_i^C\}$, $\forall i \in \mathcal{I} \setminus \{N-1\}$, where $\psi_{i,b} : (x_{i+1}, F_i') \mapsto \mathbb{R}$ is a basis function. Using the vector of coefficients $\theta_i := \{\theta_{i,1}, \ldots, \theta_{i,B_i^C}\}$, a CVFA is defined as

$$\hat{C}_i(x_{i+1}, F_i'; \theta_i) := \sum_{b=1}^{B_i^C} \psi_{i,b}(x_{i+1}, F_i')\theta_{i,b} =: (\Psi_i \theta_i)(x_{i+1}, F_i').$$

Thus, the problem of determining a VFA or a CVFA in stage i reduces to computing the vector of weights β_i or θ_i, respectively.

We specify our VFAs using a class of basis functions $\phi_{i,b}(x_i, F_i)$ such that

$$\phi_{i,b}(x_i, F_i) = \phi_{i,b}^{EN}(x_i)\phi_{i,b}^{EX}(F_i),$$

for given functions $\phi_{i,b}^{EN}(x_i)$ and $\phi_{i,b}^{EX}(F_i)$ of the endogenous and exogenous parts of the MDP state, respectively (EN and EX abbreviate endogenous and exogenous, respectively). We use analogous CVFAs. When using such a VFA, the expectation $\mathbb{E}[\hat{V}_{i+1}(x_{i+1}, \tilde{F}_{i+1}; \beta_{i+1})|F_i']$ is

$$\sum_{b=1}^{B_{i+1}^V} \phi_{i+1,b}^{EN}(x_{i+1})\mathbb{E}[\phi_{i+1,b}^{EX}(\tilde{F}_{i+1})|F_i']\beta_{i+1,b}.$$

Since in this expression $\phi^{EN}_{i+1,b}(x_{i+1})$ is outside the expectation, it can be any function of x_{i+1} that is not too expensive to evaluate. We choose $\phi^{EX}_{i+1,b}(F_{i+1})$ such that each expectation $\mathbb{E}[\phi^{EX}_{i+1,b}(\tilde{F}_{i+1})|F'_i]$ can be computed in (essentially) closed form, that is,

$$\mathbb{E}[\phi^{EX}_{i+1,b}(\tilde{F}_{i+1})|F'_i] = g_{i,b}(F'_i) \tag{10}$$

for some known function $g_{i,b}(F'_i)$. We now discuss three classes of basis functions of the exogenous part of the state that satisfy (10). We use these classes in our computational analysis in §7.

The first class of such functions are all polynomials of the term structure elements. Since such elements follow a jointly log normal distribution under model (4)-(5), $\mathbb{E}[\phi^{EX}_{i+1,b}(\tilde{F}_{i+1})|F'_i]$ can be computed in essentially closed form for these functions. For example, when $i' > i$, we can use the property $\mathbb{E}[\tilde{F}_{i',j}|F_{i,j}] = F_{i,j}$ to compute expectations of functions that are linear in the term structure elements, and the property $\mathbb{E}[\tilde{F}^2_{i',j}|F_{i,j}] = F^2_{i,j}\exp\{\sum_{k\in\mathcal{K}}\int_{T_i}^{T_{i'}}\sigma^2_{j,k}(t)dt\}$ to compute expectations of quadratic functions of such elements (these properties are easy to verify). The expectations of higher order polynomials of the term structure elements also have closed form expressions.

The second class of functions that we consider is formed by the prices of call and put options on the term structure elements: $\mathbb{E}[(\tilde{F}_{i',j} - \overline{K})^+|F_{i,j}]$ and $\mathbb{E}[(\overline{K} - \tilde{F}_{i',j})^+|F_{i,j}]$, where $i' > i$, and $\overline{K} \in \mathbb{R}$ is the given strike price. These prices can be easily computed in essentially closed form under (4)-(5).

The third class of functions that we consider includes prices of spread options on term structure elements: $\mathbb{E}[(v\tilde{F}_{i',k} - \tilde{F}_{i',j} - \overline{K})^+|F_{i,j}, F_{i,k}]$, where $k > j \geq i' > i$ and v is a given constant. Since a closed form expression for this price is not available under model (4)-(5), we instead use the near-optimal lower bound on this price developed by Bjerksund and Stensland (2011), which can be computed in essentially closed form under this term structure model.

5 LSM Methods

In this section, we focus on the LSM methods for computing the weights of a VFA or CVFA defined in §4. We start by defining the dynamic programming operators associated with MDPs (1) and (3). Using them, we develop an LSMV method for pricing multiple exercise options, describe the well known LSMC method, and compare how these methods deal with the second curse of dimensionality.

Let g^V be a function defined for all $(x_{i+1}, F_{i+1}) \in \mathcal{X}_{i+1} \times \mathbb{R}^{N-i-1}$. At stage i and state (x_i, F_i) the dynamic programming operator associated with MDP (1) is defined as

$$\mathcal{H}^V_{i,x_i,F_i}(g^V) := \max_{a\in\mathcal{A}_i(x_i)} r_i(a, F_i) + \delta\mathbb{E}\left[g^V(x_i - a, \tilde{F}_{i+1})|F'_i\right]. \tag{11}$$

Next, let g^C be a function defined for all $(x_{i+2}, F'_{i+1}) \in \mathcal{X}_{i+2} \times \mathbb{R}^{N-i-2}$. In addition, define at stage $i+1$ and state (x_{i+1}, F_{i+1}) the operator

$$\hat{\mathcal{H}}^C_{i+1,x_{i+1},F_{i+1}}(g^C) := \max_{a\in\mathcal{A}_{i+1}(x_{i+1})} r_{i+1}(a, F_{i+1}) + g^C\left(x_{i+1} - a, F'_{i+1}\right). \tag{12}$$

Then, the dynamic programming operator associated with MDP (3) at stage i and state (x_{i+1}, F_i') is defined as

$$\mathcal{H}^C_{i,x_{i+1},F_i'}(g^C) := \delta \mathbb{E}\left[\hat{\mathcal{H}}^C_{i+1,x_{i+1},\tilde{F}_{i+1}}(g^C)|F_i'\right]. \tag{13}$$

Algorithm 1: LSMV

1. Initialization: Generate the set of P term structure sample paths $\{F_i^p, i \in \mathcal{I} \setminus \{0\}, p = 1, \ldots, P\}$ and set $\bar{\beta}_N := 0$.
2. For each $i = N-1$ to 1 compute the values $\mathcal{H}^V_{i,x_i,F_i^p}(\Phi_{i+1}\bar{\beta}_{i+1})$, $\forall x_i \in \mathcal{X}_i$ and $p \in \{1, \ldots, P\}$, and perform a 2-norm regression on them to determine the weights $\bar{\beta}_i$.

LSMV. At stage i, this method computes the weights $\bar{\beta}_i$, which define the VFA $\hat{V}_i(\cdot; \bar{\beta}_i)$ as $(\Phi_i\bar{\beta}_i)(x_i, F_i)$, $\forall (x_i, F_i) \in \mathcal{X}_i \times \mathbb{R}^{N-i}$. Algorithm 1 summarizes the LSMV steps. It starts by generating the set of P term structure sample paths $\{F_i^p, i \in \mathcal{I} \setminus \{0\}, p = 1, \ldots, P\}$, and initializing the stage N weight vector $\bar{\beta}_N$ to zero. Then, at each stage $i \in \mathcal{I} \setminus \{0\}$, starting from stage $i = N-1$ and moving backwards to stage 1, Algorithm 1 computes the values $\mathcal{H}^V_{i,x_i,F_i^p}(\Phi_{i+1}\bar{\beta}_{i+1})$, $\forall x_i \in \mathcal{X}_i$ and $p \in \{1, \ldots, P\}$, and performs a 2-norm regression on them to compute the coefficients $\bar{\beta}_i$ at stage i. Computing $\mathcal{H}^V_{i,x_i,F_i^p}(\Phi_{i+1}\bar{\beta}_{i+1})$ at a state (x_i, F_i^p) involves an optimization step using the known stage $i+1$ VFA $(\Phi_{i+1}\bar{\beta}_{i+1})$ (see (11)). The expectation $\mathbb{E}[(\Phi_{i+1}\bar{\beta}_{i+1})(x_i - a, \tilde{F}_{i+1})|F_i'^p]$ in $\mathcal{H}^V_{i,x_i,F_i^p}(\Phi_{i+1}\beta_{i+1})$ can be computed in essentially closed form using the basis functions discussed in §4. This feature allows us to break the second curse of dimensionality when using LSMV.

Algorithm 2: LSMC

1. Initialization: Generate the set of P term structure sample paths $\{F_i^p, i \in \mathcal{I} \setminus \{N-1\}, p = 1, \ldots, P\}$ and set $\bar{\theta}_{N-1} := 0$.
2. For each $i = N-2$ to 0 compute the values $\delta\hat{\mathcal{H}}^C_{i+1,x_{i+1},F_{i+1}^p}(\Psi_{i+1}\bar{\theta}_{i+1})$, $\forall x_{i+1} \in \mathcal{X}_{i+1}$ and $p \in \{1, \ldots, P\}$, and perform a 2-norm regression on them to determine the weights $\bar{\theta}_i$.

LSMC. At stage i, this method computes the weights $\bar{\theta}_i$, which define the CVFA $\hat{W}_i(\cdot; \bar{\theta}_i)$ as $(\Psi_i\bar{\theta}_i)(x_{i+1}, F_i')$, $\forall (x_{i+1}, F_i') \in \mathcal{X}_{i+1} \times \mathbb{R}^{N-i-1}$. We avoid a detailed description of the LSMC steps in Algorithm 2, as they are analogous to LSMV, except for the values $\delta\hat{\mathcal{H}}^C_{i+1,x_{i+1},F_{i+1}^p}(\Psi_{i+1}\bar{\theta}_{i+1})$ used for regression in Step 2. At iteration i and for each $(x_{i+1}, F_i'^p)$, $\delta\hat{\mathcal{H}}^C_{i+1,x_{i+1},F_{i+1}^p}(\Psi_{i+1}\bar{\theta}_{i+1})$ is a *single* unbiased estimate along the p-th sample path of the stage i continuation value function $\delta\mathbb{E}\left[\hat{\mathcal{H}}^C_{i+1,x_{i+1},\tilde{F}_{i+1}}(g^C)|F_i'\right]$. Thus, LSMC avoids computing expectations.

Second curse of dimensionality. The strategies used by LSMV and LSMC for handling the second curse of dimensionality (high dimensional expectations) are not interchangeable. That is, using a basis-function-based point estimate of the expectation in (11) can result in a biased estimate

of the term $\mathcal{H}^V_{i,x_i,F_i^p}(\Phi_{i+1}\bar{\beta}_{i+1})$ at stage i and state (x_i, F_i^p) because of the maximization preceding the expectation in (11). On the other hand, closed form basis-function-based approximations of $\mathcal{H}^C_{i,x_{i+1},F_i'^p}(\Psi_{i+1}\bar{\theta}_{i+1})$ are not available, in general, since the argument to the expectation in (13) includes a maximization.

6 ALP Method

In this section, we develop an ALP method for determining a VFA. ALP solves a linear program, referred to as an approximate linear program (Schweitzer and Seidmann 1985, de Farias and Van Roy 2003), to compute the weights $\bar{\beta}_i$ that define a VFA $\hat{V}_i(\cdot;\bar{\beta}_i)$ as $(\Phi_i\bar{\beta}_i)(x_i,F_i)$, $\forall (x_i,F_i) \in \mathcal{X}_i \times \mathbb{R}^{N-i}$.

We define our approximate linear program using the set of P term structure sample paths $\{F_i^p, i \in \mathcal{I}\setminus\{0\}, p = 1,\ldots,P\}$. Let c be a vector of strictly positive state relevance weights (de Farias and Van Roy 2003) defined for all $i \in \mathcal{I}$ and $(x_i,F_i) \in \mathcal{X}_i \times \mathbb{R}^{N-i}$, where the element $c_i(x_i,F_i^p)$ associated with stage i and state (x_i,F_i^p) captures the relative importance of this state in the state space. Our approximate linear program is

$$\min_{\beta} \sum_{i\in\mathcal{I}\setminus\{0\}, x_i\in\mathcal{X}_i, p\in\{1,\ldots,P\}} c_i(x_i,F_i^p)(\Phi_i\beta_i)(x_i,F_i^p) \tag{14}$$

s.t.

$$(\Phi_i\beta_i)(x_i,F_i^p) \geq r_i(a,F_i^p) + \delta\mathbb{E}\left[(\Phi_{i+1}\beta_{i+1})(x_i-a,\tilde{F}_{i+1})|F_i'^p\right],$$
$$\forall i \in \mathcal{I}\setminus\{0,N-1\}, (x_i,p) \in \mathcal{X}_i \times \{1,\ldots,P\}, a \in \mathcal{A}_i(x_i), \tag{15}$$

$$(\Phi_{N-1}\beta_{N-1})(x_{N-1},F_{N-1}^p) \geq r_{N-1}(a,F_{N-1}^p),$$
$$\forall (x_{N-1},p) \in \mathcal{X}_{N-1} \times \{1,\ldots,P\}, a \in \mathcal{A}_{N-1}(x_{N-1}). \tag{16}$$

With a slight abuse of notation of the acronym ALP, we refer to (14)-(16) as ALP. The variables of ALP are the weights β_i, $\forall i \in \mathcal{I}\setminus\{0\}$. The ALP objective function (14) and constraints (15)-(16) can be interpreted as a sample approximation of an infinite dimensional linear program that is equivalent to (1). ALP typically has a manageable number of variables, but a large number of constraints (specifically $P \cdot \sum_{i\in\mathcal{I}\setminus\{0\}, x_i\in\mathcal{X}_i} |\mathcal{A}_i(x_i)|$) that poses a challenge to computing its optimal solution. Further, it is easy to show that ALP can be unbounded for some choice of state relevance weights. This can occur because the ALP constraints (15)-(16) are expressed with respect to the set of P term structure samples while the expectation $\mathbb{E}$ in these constraints is taken with respect to the distribution of the continuous random variable $\tilde{F}_{i+1}|F_i'^p$ under the term structure model (4)-(5).

Algorithm 3 overcomes the difficulties of solving ALP exactly. This algorithm computes a feasible solution to ALP by solving N smaller linear programs (17_i)-(18_i), each denoted by ALPD$_i$, in a backward recursive fashion. In Step 1 of Algorithm 3, we consider the same set of P sample paths of the term structure used to formulate ALP and initialize the stage N VFA to zero by defining $\bar{\beta}_N := 0$. In Step 2, we solve ALPD$_i$ to compute the weights $\bar{\beta}_i$, which are the values of an optimal solution to this linear program. The objective function of ALPD$_i$ is the stage i part of the objective function of ALP. The constraints (18_i) closely resemble an equivalent reformulation of the stage

Algorithm 3: ALPD

1. Initialization: Consider the same set of term structure sample paths that are used to formulate ALP and set $\bar{\beta}_N := 0$.

2. For each $i = N-1$ to 1 solve

$$\min_{\beta_i} \sum_{x_i \in \mathcal{X}_i, p \in \{1,\ldots,P\}} c_i(x_i, F_i^p)(\Phi_i \beta_i)(x_i, F_i^p) \tag{17_i}$$

$$\text{s.t. } (\Phi_i \beta_i)(x_i, F_i^p) \geq \mathcal{H}^V_{i,x_i,F_i^p}(\Phi_{i+1}\bar{\beta}_{i+1}), \ \ \forall (x_i, p) \in \mathcal{X}_i \times \{1,\ldots,P\}, \tag{18_i}$$

   to obtain the weights $\bar{\beta}_i$.

i constraints (15) in ALP, with one important difference: The right hand sides of the constraints (18_i) are given constants since $\bar{\beta}_{i+1}$ is already known from solving (17_{i+1})-(18_{i+1}) at stage $i+1$. This is our ALPD method.

Proposition 6.1 states two important properties of the solution computed by ALPD.

Proposition 6.1. *ALPD$_i$ has a bounded optimal solution value for all $i \in \mathcal{I} \setminus \{0\}$. Further, the ALPD solution is a basic feasible solution to ALP.*

The first property in Proposition 6.1 states that each ALPD$_i$ model can be solved to optimality for any choice of state relevance weights without encountering unboundedness. The second property in this proposition implies that the ALPD solution is an optimal solution to ALP for *some* state relevance weight vector. In other words, although ALPD is a heuristic, it can be seen as optimally solving ALP for some state relevance weight vector.

Decomposing ALP over the stages in set $\mathcal{I}$ according to Algorithm 3 has computational advantages: In our numerical experiments discussed in §7, each ALPD$_i$ model has roughly 400 times fewer variables and 5,000 times fewer constraints than ALP. Our ALP method is thus substantially more efficient than directly solving ALP to optimality.

7 Computational Results

In this section, we benchmark the computational performance of LSMV, LSMC, and ALPD on crude oil swing option instances and natural gas storage option instances. The term structure in these applications is an energy forward curve.

Our experiments are based on the following computational setup: A 64 bits PowerEdge R515 with twelve AMD Opteron 4176 2.4GHz processors, with 64GB of memory, the Linux Fedora 15 operating system, and the g++ 4.6.1 20110908 (Red Hat 4.6.1-9) compiler. We use the LAPACK 3.X library with a single processor for ordinary least squares regression and CPLEX 12.2 (CPLEX 2010) with multi-threading for solving linear programs.

7.1 Price Model and Its Calibration

For our numerical experiments, we choose the parameters $\sigma_{m,k}(t)$ of the price model (4)-(5) to be right continuous piecewise constant functions during each interval $[T_i, T_{i+1})$ (Secomandi et al. 2012). That is, we set $\sigma_{m,k}(t)$ equal to the constant $\sigma_{m,k,i}$, $\forall t \in [T_i, T_{i+1})$. Under this specification, we can equivalently rewrite (4)-(5) as

$$F(t', T_m) = F(t, T_m) \exp\left[-\frac{1}{2}(t'-t)\sum_{k=1}^{K}\sigma_{m,k,i}^2 + \sqrt{t'-t}\sum_{k=1}^{K}\sigma_{m,k,i}Z_k\right], \tag{19}$$

for all $i \in \mathcal{I}$, $m \in \{i+1, \ldots, N-1\}$, $t \in [T_i, T_{i+1})$ and $t' \in (T_i, T_{i+1}]$ with $t' > t$, and with $Z := (Z_k, k = 1, \ldots, K)$ a vector of K independent standard normal random variables. We use (19) to generate forward curve sample paths by Monte Carlo simulation.

For both crude oil and natural gas, we use ten years of NYMEX futures prices, from 1997 to 2006, to estimate monthly sample variance-covariance matrices of the daily log futures price returns for each month. We perform a principal component analysis of these matrices to estimate the loading coefficients $\sigma_{m,k,i}$ (see Clewlow and Strickland 2000 §8.6 for more details). We choose the number of factors K equal to 4 and 3 for natural gas and crude oil, respectively, as these choices explain over 99% of the total observed variance in each of our monthly data sets.

7.2 Benchmark Instances

We create four 24-stage *price* instances both for crude oil and natural gas by defining the time zero forward curve, F_0, as the forward curve for these energy sources observed on the first trading date of January, April, July, and October 2006, respectively, as these are the first months of Spring, Summer, Fall, and Winter. We add risk free interest rates equal to 4.74%, 5.05%, 5.01%, and 4.87% for the Spring, Summer, Fall, and Winter natural gas instances, respectively, following Lai et al. (2010); and the minimum of these values for the crude oil instances. We refine these price instances with application specific information to create our crude oil swing and natural gas storage option instances.

We create our swing option instances by adding to each crude oil price instance the number of swing rights n, which we vary between 1 and 10 in increments of 1, and setting the swing capacity Q_i equal to 0.2 (i.e., 20% of the base load capacity q_i set equal to 1) for each of the 24 stages (we do not consider different values for Q_i as this parameter simply scales the reward function, and hence the value of the swing option). Each strike price K_i is set equal to the time zero futures price $F_{0,i}$, that is, the price at time 0 of the futures with maturity at time T_i. We thus obtain forty swing option instances.

Our storage instances are based on our natural gas price instances and follow Lai et al. (2010) for the specification of the operational parameters. In particular, we add to each such price instance a normalized storage capacity $\bar{x}$ equal to 1, and heavy (H), moderate (M), and light (L) injection and withdrawal capacity pairs as defined in Lai et al. (2010). The initial inventory x_0 is set to zero. This process results in twelve natural gas storage instances.

7.3 Basis Functions and ALPD State Relevance Weights

We define VFAs and CVFAs using a different set of basis functions for each value of the endogenous state variable. This modeling choice corresponds to specifying the basis functions $\phi_{i,b}^{EN}(x_i)$ and $\psi_{i,b}^{EN}(x_{i+1})$ defined in §4 as indicator functions. Therefore, we represent the VFA and CVFA basis function sets as $\Phi_{i,x_i}, \forall i \in \mathcal{I} \setminus \{0\}$ and $x_i \in \mathcal{X}_i$, and $\Psi_{i,x_{i+1}}, \forall i \in \mathcal{I} \setminus \{N-1\}$ and $x_{i+1} \in \mathcal{X}_{i+1}$.

We use three different choices for each set of VFA basis functions Φ_{i,x_i} to solve our swing option instances. The first choice, labeled 1, is defined by polynomials of futures prices; namely, the constant 1, the linear and quadratic terms $F_{i,j}$ and $F_{i,j}^2, \forall j \in \{i, \ldots, N-1\}$, and the bilinear terms $F_{i,j}F_{i,k}, \forall k, j \in \{i, \ldots, \min\{i+4, N-1\}\}$ with $k > j$ (the number of bilinear terms to use was determined by experimentation). The second choice, labeled 2, includes the constant 1, and the call and put option prices $\mathbb{E}[(\tilde{F}_{k,k} - F_{0,k})^+|F_{i,k}]$ and $\mathbb{E}[(F_{0,k} - \tilde{F}_{k,k})^+|F_{i,k}]$, $\forall k \in \{i, \ldots, N-1\}$. This choice is based on the observation that the reward function and the optimal value function boundary case with $n = N$ can be modeled using pairs of call and put option prices. The third choice, labeled 3, is the union of the basis functions used in the two previous cases. We also use three specifications for the set of CVFA basis functions $\Psi_{i,x_{i+1}}$, labeled 1, 2, and 3. They are defined analogously to the basis functions of each set Φ_{i,x_i}, except that they do not depend on the spot price at stage i, $F_{i,i}$, since the CVFA is a function of $F_i' \equiv \{F_{i,i+1}, \ldots, F_{i,N-1}\}$ rather than $F_i \equiv \{F_{i,i}, F_{i,i+1}, \ldots, F_{i,N-1}\}$.

To solve our storage option instances, we use two types of VFA and CVFA basis function sets. The first such VFA set, labeled 1, is identical to the one labeled 1 for the swing option case. The second VFA basis function set, labeled 2, is defined by the constant 1, the spot price $F_{i,i}$, and the prices of spread options $\mathbb{E}[(\delta\tilde{F}_{k,k+1}^W - \tilde{F}_{k,k}^I)^+|F_{i,k}, F_{i,k+1}]$, where $k \in \{i, \ldots, N-2\}$, $F_{k,k+1}^W := \alpha^W F_{k,k+1} - \varsigma^W$, and $F_{k,k}^I := \alpha^I F_{k,k} + \varsigma^I$. This choice is based on the finding by Secomandi (2012) that the optimal value function of MDP (1) applied to storage with unitary injection and withdrawal loss coefficients, zero injection and withdrawal marginal costs, and injection and withdrawal capacities equal to the space ($\alpha^I = \alpha^W = 1$, $\varsigma^I = \varsigma^W = 0$, and $|\underline{a}| = \overline{a} = \overline{x}$) is of this form. We also use two analogous specifications, labeled 1 and 2, for the CVFA basis function sets.

When formulating ALPD, we use the state relevance weights $c_i(x_i, F_i^p) = 1/(|\mathcal{X}_i|P)$. With these state relevance weights, the ALPD objective function (17_i) can be written as

$$[1/(|\mathcal{X}_i|P)] \sum_{x_i \in \mathcal{X}_i, p \in \{1,\ldots,P\}} (\Phi_{i,x_i}\beta_{i,x_i})(F_i^p),$$

which is a sample approximation of $(1/|\mathcal{X}_i|)\mathbb{E}\left[\sum_{x_i \in \mathcal{X}_i}(\Phi_{i,x_i}\beta_{i,x_i})(\tilde{F}_i)|F_0'\right]$.

7.4 Swing Option

In this subsection, we compare the performance of the LSMV, LSMC, and ALPD methods on our swing option instances. Recall that P denotes the number of samples used to compute VFAs and CVFAs and W denotes the number of samples used to estimate lower and upper bounds in Monte Carlo simulation. We estimate greedy lower and dual upper bounds via Monte Carlo simulation using 100,000 forward curve sample paths; that is, W is equal to 100,000 (see §3 for details).

When LSMV, LSMC, and ALPD are combined with our three definitions of Φ_{i,x_i} and $\Psi_{i,x_{i+1}}$, we obtain nine versions of these methods, labeled as follows: LSMV1, LSMV2, and LSMV3; LSMC1, LSMC2, and LSMC3; and ALPD1, ALPD2, and ALPD3.

Upper Bounds. We compare our upper bound estimates obtained with the number of regression samples P set equal to 1,000 – further increasing the value of P leads to looser ALPD upper bound estimates. Our error analysis in §8 sheds light on this ALPD behavior. As discussed at the end of this subsection, estimating upper bounds using LSMC, with penalties defined by (9), requires two orders of magnitude more CPU time than estimating upper bounds using LSMV or ALPD. Therefore, for the purpose of comparing the quality of upper bound estimates, we only report the estimated LSMC3 upper bound, which is the tightest of the estimated LSMC upper bounds.

Figure 1: Comparison of the estimated upper bounds on the swing option instances as percentages of the LSMV3 upper bound estimates for $n = 3$ and $P = 1,000$.

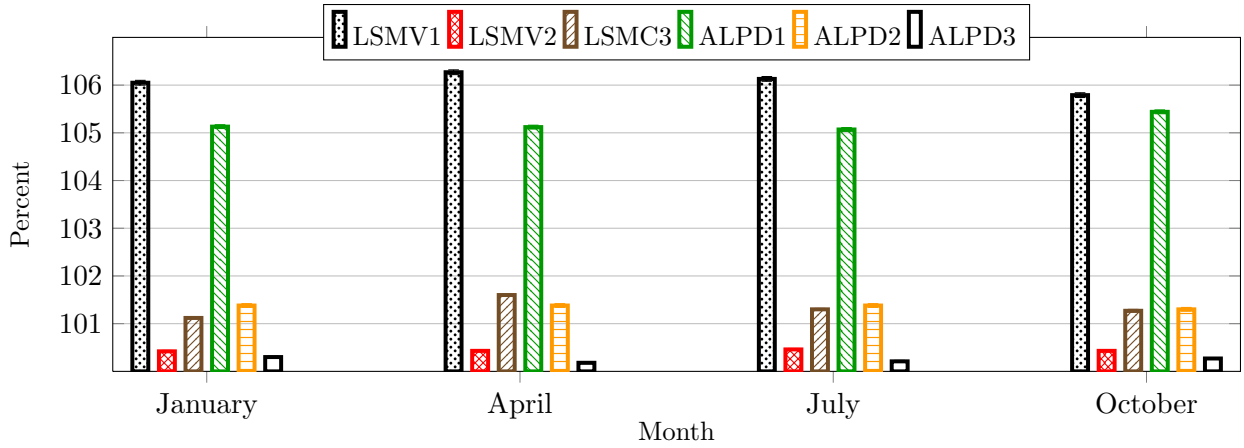

Figure 1 displays the estimated upper bounds and their standard errors (the latter are barely visible as they are in the order of 0.01%) as percentages of the LSMV3 upper bound estimates (the standard error of the LSMV3 upper bound estimate is at most 0.05% of this estimated bound) for our swing option instances with three exercise rights ($n = 3$). These results show that option based VFAs (labeled 2) deliver tighter upper bound estimates than polynomial based VFAs (labeled 1), and the combination of these two VFA types results in even tighter upper bound estimates. Moreover, LSMV3 yields the tightest upper bound estimate on each instance. The LSMC3 and ALPD3 estimated upper bounds are looser than the LSMV3 estimated upper bound by at most 1.60% and 0.30%, respectively. The relative performance of these estimated bounds is similar for other values of the number of exercise rights.

As the value of P increases from 1,000 to 10,000, the LSMV3 upper bound estimates do not change beyond standard error; the LSMC3 upper bound estimates improve to within standard error of their respective LSMV3 upper bound estimates with P equal to 1,000; and all the ALPD upper bound estimates worsen. Hence, LSMV3 attains stable upper bound estimates, which are also the tightest upper bound estimates, using an order of magnitude fewer samples than LSMC3.

Lower Bounds. We compare the lower bounding performance of our methods executed with

P equal to 1,000, since the greedy lower bounds estimated using ALPD deteriorate as the value of P is increased further. Our error analysis in §8 sheds light on this ALPD behavior. Across all our swing option instances, the standard errors of all the greedy lower bound estimates are between 0.15% and 0.19% of these estimates. We normalize all the reported lower bound estimates by their respective estimated LSMV3 upper bounds.

Table 1 reports our estimated lower bounds for the instances with three exercise rights ($n = 3$). The estimated LSMV2 and LSMV3 lower bounds are within standard error of each other and dominate the LSMV1 lower bound estimate; the LSMC2 lower bound estimates dominate the ones from both LSMC1 and LSMC3; and the ALPD3 lower bound estimates dominate the ones from both ALPD1 and ALPD2. LSMV2 and LSMV3 estimate the tightest lower bounds across the nine methods, with a maximum suboptimality gap of 0.13%, which is less than the standard error of these estimates. It is interesting that the LSMC2 estimated lower bounds dominate the ones obtained with LSMC3 on all these instances when the number of samples P is equal to 1,000, even though the basis functions used by LSMC3 include those used by LSMC2. This is a convergence issue: We have verified that the LSMC3 estimated lower bounds obtained with P equal to 10,000 are within standard error of the LSMV3 lower bound estimated with P equal to 1,000. In contrast to the suboptimality of the LSMV and LSMC methods, the suboptimality of the ALPD method can be substantial.

Across all of our swing option instances, that is, also considering the remaining values of n, the lower bounds estimated by LSMV3 and LSMC3 have maximum suboptimality gaps of 0.5% and 4%, respectively. However, when P is set equal to $10,000$ the LSMC3 lower bound estimates are within standard error of the LSMV3 lower bound estimates. Thus, LSMV3 converges and estimates the tightest lower bounds requiring an order of magnitude fewer samples than LSMC3.

Table 1: Comparison of the estimated greedy lower bounds on the swing option instances as percentages of the LSMV3 upper bound estimates for $n = 3$ and $P = 1,000$.

Month	Version	LSMV	LSMC	ALPD	Month	Version	LSMV	LSMC	ALPD
January	1	98.28	97.72	90.21	July	1	99.52	97.59	89.97
	2	99.92	99.34	93.55		2	99.95	99.18	93.31
	3	100.01	98.78	98.40		3	100.03	98.61	98.67
April	1	98.64	97.91	90.54	October	1	98.18	97.06	89.87
	2	99.96	99.20	93.87		2	99.87	99.08	93.17
	3	100.09	98.65	98.92		3	99.97	98.38	98.49

CPU Times. Table 2 reports the average run times to compute a VFA or a CVFA, estimate a greedy lower bound, and estimate a dual upper bound when using LSMV3 and ALPD3 with P equal to 1,000, and LSMC3 with P equal to 10,000 (recall that LSMC requires 10,000 regression samples to compute a CVFA that delivers stable lower bounds). To illustrate the dependence of the required computational effort on the value of the parameter n, Table 2 reports results for both $n = 1$ and $n = 10$. We average over the results corresponding to the January, April, July, and October instances as the computational times needed to solve these instances are essentially

the same. Increasing the value of the parameter n does not affect the observed CPU times in a significant manner. The LSMV and LSMC methods compute a VFA and a CVFA, respectively, in a faster fashion than the ALPD method computes a VFA. Moreover, on average, LSMV takes between computes in 0.1 and 0.6 CPU seconds to compute a VFA that yields stable lower and upper bound estimates, while LSMC requires between 3 and 8 CPU seconds to compute an analogous CVFA. Thus, on average, the convergence of LSMV is one order of magnitude faster than the one of LSMC.

Table 2: Average CPU times for computing a VFA or a CVFA and estimating lower and upper bounds on a subset of the swing option instances with $P = 1,000$ for ALPD and LSMV and $P = 10,000$ for LSMC.

		VFA/CVFA			Lower Bound			Upper Bound		
n	Version	LSMV	LSMC	ALPD	LSMV	LSMC	ALPD	LSMV	LSMC	ALPD
1	1	0.11	3.63	5.28	0.86	0.73	0.87	0.82	326.64	0.71
	2	0.38	6.12	4.36	14.61	13.78	14.68	13.66	1,642.79	13.53
	3	0.52	7.19	9.01	15.07	13.49	14.94	14.06	1,699.40	14.18
10	1	0.15	4.12	17.85	1.84	0.77	1.83	2.91	838.52	2.79
	2	0.42	6.57	15.14	15.14	13.99	15.08	15.80	2,119.33	15.69
	3	0.58	7.99	28.30	16.40	14.50	16.48	16.82	2,424.97	16.62

As expected, the run times needed to estimate lower and upper bounds using LSMV and ALPD are comparable, since these methods use the same bounding procedures once a VFA is specified. On average, it takes LSMC at most 2 additional CPU seconds to estimate greedy lower bounds given a CVFA compared to the CPU time incurred by both LSMV and ALPD to estimate lower bounds given a VFA. Since lower bound estimation requires computing a VFA or a CVFA and estimating a lower bound, the LSMV CPU times are 1.32-4.49 times smaller on average than the LSMC CPU times for lower bound estimation. The CPU times needed to estimate valid upper bounds using LSMC are more than two orders of magnitude larger than the ones taken by both LSMV and ALPD.

These findings suggest that LSMV estimates its tightest lower bounds in a faster fashion than LSMC and that VFAs have a substantial computational advantage over CVFAs when estimating upper bounds.

7.5 Storage Option

In this subsection, we discuss the performance of the LSMV, LSMC, and ALPD methods applied to our storage option instances. We estimate lower and upper bounds using 100,000 forward curve sample paths by Monte Carlo simulation, that is, W is equal to 100,000.

Upper Bounds. We compare the upper bounds estimated with the number of samples P equal to 1,000, as the upper bounds estimated by ALPD worsen when using more samples (see §8 for a possible explanation). LSMC requires at least 5 hours to compute a dual upper bound using the penalties (9). Thus, LSMC is not a practical method for upper bound estimation in this application. However, for the purpose of comparing upper bound quality, we only report the estimated LSMC1 upper bounds, which are the tightest of all the estimated LSMC upper bounds.

Figure 2: Comparison of the estimated upper bounds on the storage option instances as percentages of the estimated LSMV1 upper bounds for P =1,000.

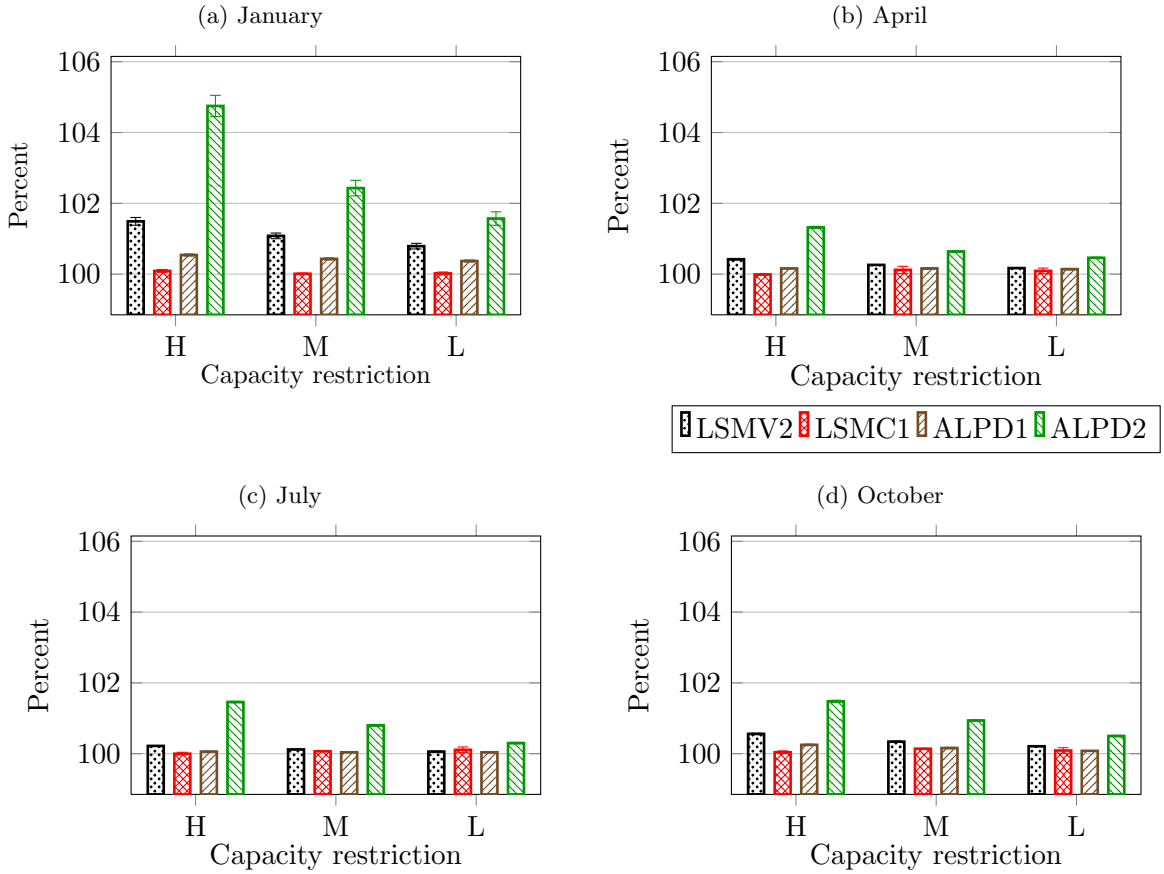

Figure 2 displays the estimated upper bounds and their standard errors as percentages of the LSMV1 upper bound estimates (the standard errors of the estimated LSMV1 upper bounds vary between 0.01% and 0.08% of these estimated bounds). The vertical axes in the graphs of Figure 2 start below 100 since some of the LSMV1 and LSMC1 upper bound estimates minus their standard errors are marginally smaller than 100% (by at most 0.03%). LSMV1 and LSMC1 estimate the tightest upper bounds on almost all the instances (these estimates are within standard error of each other). ALPD1 is a close second with its largest estimated upper bound being 0.5% of the LSMV1 estimated upper bounds. The upper bounds estimated by LSMV1 and ALPD1 dominate the ones estimated by LSMV2 and ALPD2, respectively. Hence, basis functions based on polynomials (labeled 1) yield better upper bound estimates than basis functions based on spread options (labeled 2) on these instances with P equal to 1,000. As also observed on our swing option instances, increasing the value of P does not improve the upper bound estimated by LSMV1 beyond standard error.

Lower Bounds. We compare our estimated greedy lower bounds with the number of samples P equal to 1,000, as the ALPD lower bounds worsen for larger values of P (see §8 for a possible

Table 3: Comparison of the estimated greedy lower bounds on the storage option instances as percentages of the estimated LSMV1 upper bounds for P =1,000.

(a) January

Capacity	Version	LSMV	LSMC	ALPD
H	1	99.54	97.12	99.06
	2	98.93	97.81	73.70
M	1	99.41	95.49	98.91
	2	98.90	96.97	84.20
L	1	99.34	94.68	98.95
	2	99.00	96.70	90.48

(b) April

Capacity	Version	LSMV	LSMC	ALPD
H	1	99.91	99.51	99.79
	2	99.62	99.41	93.30
M	1	99.84	98.77	99.57
	2	99.61	99.09	95.64
L	1	99.86	98.49	99.60
	2	99.74	99.07	96.64

(c) July

Capacity	Version	LSMV	LSMC	ALPD
H	1	99.99	99.57	99.95
	2	99.84	99.59	95.22
M	1	100.04	99.36	99.97
	2	99.92	99.56	98.19
L	1	100.10	99.16	100.02
	2	100.05	99.52	99.17

(d) October

Capacity	Version	LSMV	LSMC	ALPD
H	1	99.81	98.94	99.55
	2	99.47	99.10	86.12
M	1	99.82	98.56	99.55
	2	99.54	98.98	94.62
L	1	99.84	98.51	99.70
	2	99.70	99.01	96.48

explanation). All the lower bound estimates reported in the rest of this subsection are normalized by the LSMV1 upper bounds estimated with P equal to 1,000. Table 3 reports these estimated bounds (their standard errors vary between 0.11% and 0.34%). LSMV1 estimates the tightest lower bounds, with a maximum suboptimality gap of 0.66%. LSMC2 and ALPD1 estimate the tightest LSMC and ALPD lower bounds, respectively, each with a maximum suboptimality of 3.30% and 1.09%. The estimated ALPD1 lower bounds dominate the LSMC1 ones while the estimated LSMC2 lower bounds dominate the ALPD2 ones. Further, the lower bounds estimated by LSMV1 and ALPD1, respectively, dominate the LSMV2 and ALPD2 ones.

Figure 3 compares the convergence of the estimated LSMV1, LSMC1, and LSMC2 lower bounds on the January instances with heavy or light capacity restrictions (the convergence trend for the other instances is similar). The LSMV1 lower bound estimates converge to a stable value with 1,000 regression samples, while the LSMC1 lower bound estimates require about 10,000 regression samples to converge. Interestingly, even though the lower bounds estimated by LSMC2 dominate the LSMC1 ones on almost all the instances with 1,000 such samples, the reverse is true when the number of these samples is increased to 10,000. The estimated LSMV1 lower bounds and the stable estimated LSMC1 lower bounds are within standard error of each other. However, similar to our results obtained on the swing option instances, the best LSMV lower bound estimates converge to a stable value when using 10 times fewer regression samples than those required to estimate the best LSMC lower bounds. Further, as observed when discussing our swing option instances, the estimated ALPD lower bounds can be substantially inferior compared to the lower bounds estimated by LSMV or LSMC.

Figure 3: Convergence of the greedy lower bounds estimated by LSMV1, LSMC1, and LSMC2 as percentages of the estimated LSMV1 upper bounds on the January storage option instances with heavy and light capacity restrictions.

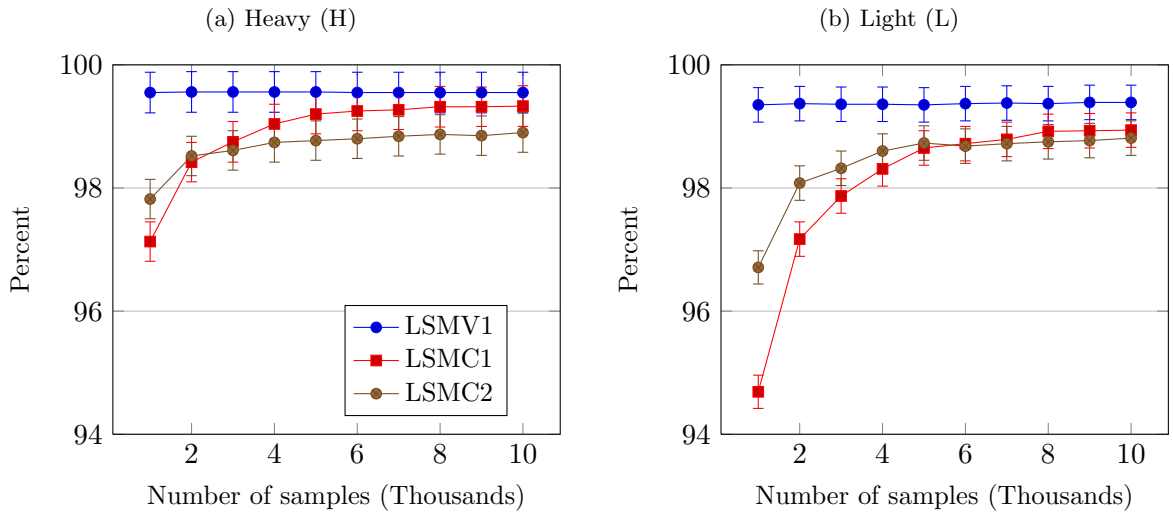

Table 4: Average CPU times for computing a VFA or a CVFA and estimating lower and upper bounds on the storage option instances with $P = 1,000$ for ALPD and LSMV and $P = 10,000$ for LSMC.

		VFA/CVFA			Lower Bound			Upper Bound		
Capacity	Version	LSMV	LSMC	ALPD	LSMV	LSMC	ALPD	LSMV	LSMC	ALPD
H	1	0.45	5.40	28.39	3.53	2.29	3.78	13.62	20,149.28	14.48
	2	0.53	6.15	7.56	16.05	14.42	16.30	24.92	27,608.48	28.33
M	1	0.70	8.20	29.05	4.17	3.31	4.55	22.35	56,550.55	23.10
	2	0.70	8.16	7.38	16.68	15.15	16.84	33.57	81,329.48	34.13
L	1	0.87	9.99	29.38	4.65	4.14	5.11	27.63	87,828.51	29.71
	2	0.80	9.36	6.79	17.13	15.65	17.23	38.77	128,179.65	38.70

CPU Times. Table 4 reports the average run times over all our storage instances required to compute a VFA or a CVFA and estimate greedy lower and upper bounds using LSMV and ALPD with P equal to 1,000 and LSMC with P equal to 10,000. The ranges of the average run times required to compute a VFA using LSMV and ALPD, respectively, are 0.45-0.87 and 6.79-29.38 CPU seconds. The average run time taken by LSMC to compute a CVFA ranges between 5.40 and 9.99 CPU seconds. Therefore, computing a functional approximation that delivers stable lower and upper bounds with LSMV is at least 10 times faster than with LSMC and 14-60 times faster than with ALPD. The run times required to estimate lower bounds given a VFA or CVFA are comparable for all the three algorithms, with LSMC being faster than LSMV by at most 2 seconds. Since the lower bound estimation also requires computing a VFA or a CVFA, on average LSMV is 1.24 to 2.55 times faster than LSMC. The CPU time taken by LSMV and ALPD to estimate upper bounds are comparable, since these methods use the same bounding procedure once a VFA is computed.

LSMC requires at least three orders of magnitude more CPU seconds than both LSMV and ALPD to estimate valid dual upper bounds.

As concluded when discussing our results obtained on our swing option instances, these findings suggest that LSMV estimates its tightest lower bounds in a faster manner than LSMC and that VFAs have a substantial computational advantage over CVFAs when estimating upper bounds.

8 Error Bound Analysis

In this section, we analyze the LSM and ALP methods to provide some theoretical support for their numerical upper and lower bounding performance discussed in §7. We derive ∞-norm worst case error bounds on the VFAs computed by LSMV and a more analytically tractable version of ALPD, and of the CVFA computed by LSMC. The premise behind our analysis is that VFAs/CVFAs with smaller error bounds should yield better lower and upper bound estimates.

We first state the assumptions that support our analysis and then discuss our error bounds.

Assumptions. We focus on VFAs and CVFAs with respect to sampled versions of the MDPs (1) and (3) constructed using P sample paths in the set $\{F_i^p, i \in \mathcal{I}, p = 1, \ldots, P\}$ of the term structure F_i starting from F_0. Denote by $\mathbb{E}^s$ expectation with respect to a probability distribution on the sampled term structures (the superscript s denotes a sampled version). Let $\mathcal{H}_i^{V,s}$ and $\mathcal{H}_i^{C,s}$ be the stage i dynamic programming operators defined analogously to (11) and (13), except that expectation $\mathbb{E}$ appearing in (11) and (13) is replaced by $\mathbb{E}^s$. At stage i, let $V_i^s(x_i, F_i^p)$ be the optimal value function in state (x_i, F_i^p) and $C_i^s(x_{i+1}, F_i'^p)$ be the optimal continuation value function in state $(x_{i+1}, F_i'^p)$ of these sampled MDPs. The sampled finite state value function MDP can be stated as $V_i^s(x_i, F_i^p) = \mathcal{H}_{i,x_i,F_i^p}^{V,s}(V_{i+1}^s)$ or, equivalently, as

$$V_i^s(x_i, F_i^p) = \max_{a \in \mathcal{A}_i(x_i)} r_i(a, F_i^p) + \delta \mathbb{E}^s \left[V_{i+1}^s \left(x_i - a, \tilde{F}_{i+1} \right) | F_i'^p \right], \tag{20}$$

$\forall i \in \mathcal{I}$ and $(x_i, p) \in \mathcal{X}_i \times \{1, \ldots, P\}$, with boundary conditions $V_N^s(x_N, F_N^p) := 0, \forall x_N \in \mathcal{X}_N$. The corresponding sampled finite state continuation value function MDP can be stated as $C_i^s(x_{i+1}, F_i'^p) = \mathcal{H}_{i,x_{i+1},F_i'^p}^{C,s}(C_{i+1}^s)$ or, equivalently, as

$$C_i^s(x_{i+1}, F_i'^p) = \delta \mathbb{E}^s \left[\max_{a \in \mathcal{A}_{i+1}(x_{i+1})} r_{i+1}(a, \tilde{F}_{i+1}) + C_{i+1}^s \left(x_{i+1} - a, \tilde{F}'_{i+1} \right) | F_i'^p \right], \tag{21}$$

$\forall i \in \mathcal{I} \backslash \{N-1\}$ and $(x_{i+1}, p) \in \mathcal{X}_{i+1} \times \{1, \ldots, P\}$, with boundary conditions $C_{N-1}^s(x_N, F'^p_{N-1}) := 0$, $\forall x_N \in \mathcal{X}_N$.

Let $\|g(\cdot)\|_2 := \left(\sum_{d \in \mathcal{D}} (g(d))^2 \right)^{1/2}$ and $\|g(\cdot)\|_{1,c} := \sum_{d \in \mathcal{D}} |g(d)|$ denote the 2-norm and 1,c-norm, respectively, of a function g defined on the finite set $\mathcal{D}$. To facilitate our error analysis we view LSMV, LSMC, and ALPD as regression based methods. A regression interpretation of LSMV and LSMC is obvious since they perform a 2-norm regression for computing a VFA and a CVFA, respectively. On the contrary, such a regression interpretation is not immediate for the ALPD approach outlined in Algorithm 3. Proposition 8.1 brings out the implicit regression step in ALPD

by reformulating the linear program ALPD$_i$ as a constrained $1,c$-norm regression problem. This result is similar to Lemma 1 in de Farias and Van Roy (2003).

Proposition 8.1. *The optimal solution to ALPD$_i$ is also optimal to the math program*

$$\min_{\beta_i} \left\| (\Phi_i\beta_i)(x_i, F_i^p) - \mathcal{H}^V_{i,x_i,F_i^p}(\Phi_{i+1}\bar{\beta}_{i+1}) \right\|_{1,c} \tag{22$_i$}$$

$$s.t.\ (\Phi_i\beta_i)(x_i, F_i^p) \geq \mathcal{H}^V_{i,x_i,F_i^p}(\Phi_{i+1}\bar{\beta}_{i+1}),\ \ \forall (x_i, p) \in \mathcal{X}_i \times \{1, \ldots, P\}, \tag{23$_i$}$$

where $\bar{\beta}_{i+1}$ in (22$_i$)-(23$_i$) is the optimal solution to (22$_{i+1}$)-(23$_{i+1}$) if $i \in \mathcal{I} \setminus \{0, N-1\}$ and is equal to a vector of zeros if $i = N-1$.

The formulation (22$_i$)-(23$_i$) uses a different norm than the norm used by LSMV and LSMC. This aspect complicates our comparison of the ALPD VFA error and the analogous errors for LSMV and LSMC. We address this issue by analyzing a modified version of ALPD, referred to as 2ALPD, which uses a 2-norm instead of the 1,c-norm in (22$_i$) and the Bellman operator $\mathcal{H}^{V,s}$ in lieu of $\mathcal{H}^V$ in (22$_i$)-(23$_i$). The latter change is for consistency with the finite sampled MDPs considered in this section. Specifically, 2ALPD solves the following set of convex programs:

$$\min_{\beta_i} \left\| (\Phi_i\beta_i)(x_i, F_i^p) - \mathcal{H}^{V,s}_{i,x_i,F_i^p}(\Phi_{i+1}\bar{\beta}_{i+1}) \right\|_2 \tag{24$_i$}$$

$$\text{s.t. } (\Phi_i\beta_i)(x_i, F_i^p) \geq \mathcal{H}^{V,s}_{i,x_i,F_i^p}(\Phi_{i+1}\bar{\beta}_{i+1}),\ \ \forall (x_i, p) \in \mathcal{X}_i \times \{1, \ldots, P\}, \tag{25$_i$}$$

$\forall i \in \mathcal{I} \setminus \{0\}$ and with $\bar{\beta}_N := 0$. In (24$_i$)-(25$_i$), β_i is the vector of decision variables, and $\bar{\beta}_{i+1}$ is the optimal solution to (24$_{i+1}$)-(25$_{i+1}$).

For ease of analysis, we make a standard assumption that guarantees a unique solution to the 2-norm regression problem $\min_{\beta} \left\| (\hat{\Phi}\beta)(\cdot) - g(\cdot) \right\|_2$, where g is a function defined on the finite set $\mathcal{D}$ with cardinality denoted by $|\mathcal{D}|$; $\hat{\Phi}$ is a set of $\hat{B}$ basis functions $\{\hat{\phi}_1, \ldots, \hat{\phi}_{\hat{B}}\}$; and β is the vector of unknown regression coefficients. Define Y as the $|\mathcal{D}| \times \hat{B}$ regression matrix whose element in position (d, b) is $Y_{d,b} := \hat{\phi}_b(d)$, $\forall (d, b) \in \mathcal{D} \times \{1, \ldots, \hat{B}\}$. In this subsection, we assume that each regression matrix Y associated with a regression problem has full column rank. Under this assumption, the unique optimal solution to the 2-norm regression problem $\min_{\beta} \left\| (\hat{\Phi}\beta)(\cdot) - g(\cdot) \right\|_2$ is $\bar{\beta}_i = (Y^T Y)^{-1} Y^T g$. We define the 2-norm projection operator Π_2 applied to g as $\Pi_2 g := \hat{\Phi}\bar{\beta}$.

Analysis. We denote the solutions obtained by the LSMV, 2ALPD, and LSMC methods by $\bar{\beta}_i^{LV}$, $\bar{\beta}_i^{LP}$, and $\bar{\theta}_i^{LC}$, respectively. Define by $\|g(\cdot)\|_\infty := \max_{d \in \mathcal{D}} |g(d)|$ the ∞-norm operating on a function g with support on the finite set $\mathcal{D}$. Table 5 lists the definitions of the errors used in our investigation. We discuss each error in the ensuing analysis.

Theorem 8.2 presents bounds on the LSMV and 2ALPD VFA errors $\tilde{e}_i^{LV}$ and $\tilde{e}_i^{LP}$ at each stage $i \in \mathcal{I} \setminus \{0\}$. These bounds depend on the 2-norm regression error e_i^* that results when using basis functions to approximate the value function V_i^s. In addition, the 2ALPD error $\tilde{e}_i^{LP}$ depends on $\bar{e}_i^{LP}$, the error resulting from the 2ALPD constraints requiring the stage i VFA to be an upper bound on the value function $\mathcal{H}^{V,s}_{i,x_i,F_i^p}(\Phi_{i+1}\bar{\beta}_{i+1})$ induced by the stage $i+1$ VFA at each of the sampled

Table 5: Error definitions.

Category	Name and Definition
Regression	$e_i^* := \min_{\beta_i} \left\| \hat{V}_i(\cdot;\beta_i) - V_i^s(\cdot) \right\|_2 \equiv \|\Pi_2 V_i^s(\cdot) - V_i^s(\cdot)\|_2$
	$e_i^{**} = \min_{\beta_i} \left\| \hat{C}_i(\cdot;\theta_i) - C_i^s(\cdot) \right\|_2 \equiv \|\Pi_2 C_i^s(\cdot) - C_i^s(\cdot)\|_2$
LSMV	$\tilde{e}_i^{LV} := \left\| \hat{V}_i(\cdot;\bar{\beta}_i^{LV}) - V_i^s(\cdot) \right\|_\infty$
2ALPD	$\tilde{e}_i^{LP} := \left\| \hat{V}_i(\cdot;\bar{\beta}_i^{LP}) - V_i^s(\cdot) \right\|_\infty$
	$\bar{e}_i^{LP} := \left\| \hat{V}_i(\cdot;\bar{\beta}_i^{LP}) - \Pi_2 \mathcal{H}_{i,\cdot,\cdot}^{V,s} \left(\hat{V}_{i+1}(\cdot;\bar{\beta}_{i+1}^{LP}) \right) \right\|_\infty$
LSMC	$\tilde{e}_i^{LC} := \left\| \hat{C}_i(\cdot;\bar{\theta}_i^{LC}) - C_i^s(\cdot) \right\|_\infty$
	$\bar{e}_i^{LC} := \left\| \hat{C}_i(\cdot;\bar{\theta}_i^{LC}) - \Pi_2 \mathcal{H}_{i,\cdot,\cdot}^{C,s} \left(\hat{C}_{i+1}(\cdot;\bar{\theta}_{i+1}^{LC}) \right) \right\|_\infty$

states. This interpretation relies on two considerations: (i) $\Pi_2 \mathcal{H}_{i,\cdot,\cdot}^{V,s} \left(\hat{V}_{i+1}(\cdot;\bar{\beta}_{i+1}^{LP}) \right)$ is the result of an unconstrained 2-norm regression over the estimates $\mathcal{H}_{i,x_i,F_i^p}^{V,s}(\Phi_{i+1}\bar{\beta}_{i+1})$ at each sampled state (x_i, F_i^p); (ii) ALPD$_i$ performing a constrained 2-norm regression over these values in the presence of constraints (25_i).

Theorem 8.2. *It holds that* $\tilde{e}_i^{LV} \leq \sum_{j=i}^{N-1} \delta^{j-i} e_j^*$ *and* $\tilde{e}_i^{LP} \leq \sum_{j=i}^{N-1} \delta^{j-i}(e_j^* + \bar{e}_j^{LP})$, $\forall i \in \mathcal{I} \setminus \{0\}$.

Theorem 8.2 shows that the LSMV and ALPD errors in stage i respectively, are no more than the discounted sum of the regression errors e_j^* and the discounted sum of the regressions errors e_j^* *and* the errors $\bar{e}_j^{LP}$ for the remaining stages. Thus, our bound on the stage i LSMV error is smaller than the one on the analogous 2ALPD error. Recall from §7.3 that we use the state relevance weights $c(x_i, F_i^p) = 1/(|\mathcal{X}_i|P)$ to obtain our numerical results with ALPD. Since these weights are strictly positive and equal for all the sampled states (x_i, F_i^p), the optimal ALPD solution under this 1,c-norm is optimal if we use any strictly positive and equal state relevance weights in (22_i), in particular, a 1-norm. In addition, a 2-norm discourages large deviations more than a 1-norm. Therefore, the VFA associated with an optimal solution to 2ALPD should typically have a smaller ∞-norm error compared to the VFA associated with an optimal solution to ALPD. These arguments and Theorem 8.2 *suggest* a smaller worst case error for the LSMV VFA compared to the one for ALPD. This statement thus supports the superior numerical bounding performance of LSMV over ALPD discussed in §7.

Theorem 8.3 bounds the error $\tilde{e}_i^{LC}$ that arises from approximating the continuation value function C_i^s obtained by LSMC. We denote by e_i^{**} the regression error incurred when approximating C_i^s using basis functions. In addition, we denote by $\bar{e}_i^{LC}$ the error incurred from constructing a CVFA by regressing over the estimates $\delta\hat{\mathcal{H}}_{i+1,x_{i+1},F_{i+1}^p}^{C,s} \left(\hat{C}_{i+1}(\cdot,\cdot;\bar{\theta}_{i+1}^{LC}) \right)$ of the stage i continuation value function induced by the stage i+1 CVFA $\mathcal{H}_{i,\cdot,\cdot}^{C,s} \left(\hat{C}_{i+1}(\cdot,\cdot;\bar{\theta}_{i+1}^{LC}) \right)$, instead of regressing over the evaluations of this induced continuation value function at the sampled states. This is clearly an approximation since $\delta\hat{\mathcal{H}}_{i+1,x_{i+1},F_{i+1}^p}^{C,s} \left(\hat{C}_{i+1}(\cdot,\cdot;\bar{\theta}_{i+1}^{LC}) \right)$ is a single sample estimate of $\mathcal{H}_{i,x_{i+1},F_i'^p}^{C,s} \left(\hat{C}_{i+1}(\cdot,\cdot;\bar{\theta}_{i+1}^{LC}) \right)$ (see

(13)).

Theorem 8.3. *It holds that* $\tilde{e}_i^{LC} \leq \sum_{j=i}^{N-2} \delta^{j-i}(e_j^{**} + \bar{e}_j^{LC})$, $\forall i \in \mathcal{I} \setminus \{N-1\}$.

Theorem 8.3 shows that the LSMC error in stage i, $\tilde{e}_i^{LC}$, is no more than the discounted sum of the regression errors e_j^{**} *and* the errors $\bar{e}_j^{LC}$ for the remaining stages. The error term $\bar{e}_j^{LC}$ arises from approximating the function $\mathcal{H}_{i,\cdot,\cdot}^{C,s}\left(\hat{C}_{i+1}(\cdot,\cdot;\bar{\theta}_{i+1}^{LC})\right)$ using a cross-sectional regression of basis functions over the sample estimates $\delta\hat{\mathcal{H}}_{i+1,x_{i+1},F_{i+1}^p}^{C,s}\left(\hat{C}_{i+1}(\cdot,\cdot;\bar{\theta}_{i+1}^{LC})\right)$. This error should thus decrease when increasing the number of regression samples P. This supports the improvements observed in §7 of the LSMC greedy lower bound estimates obtained when increasing the value of P. The deterioration of the lower and upper bounding performance of ALPD when increasing the value of P observed in §7 stands in stark contrast to this improvement. If the chosen basis functions do not span the value function V_i^s, which is typically the case, this observed behavior can be attributed to the increase in the error $\bar{e}_j^{LP}$ that should result from using a larger number of samples (this statement relies on the 2ALPD constraints requiring the VFA to be an upper bound on $\mathcal{H}_{i,x_i,F_i^p}^{V,s}(\Phi_{i+1}\bar{\beta}_{i+1})$ at an increased number of states).

The error bounds on the VFAs determined by LSMV and ALPD stated in Theorem 8.2 are not directly comparable with the error bound on the CVFA obtained with LSMC stated in Theorem 8.3. Therefore, we focus on the estimation of a greedy lower bound (see §3) using the CVFA $C_i^I(x_{i+1}, F_i'^p; \beta_{i+1}) = \delta\mathbb{E}^s\left[\hat{V}_{i+1}(x_{i+1}, \tilde{F}_{i+1}; \beta_{i+1})|F_i'^p\right]$ induced by the VFA $\hat{V}_i(x_i, F_i^p; \beta_i)$ (the superscript I denotes induced). Corollary 8.4 presents bounds on the errors associated with the CVFAs induced by the LSMV and 2ALPD VFAs. We define these errors as $\tilde{e}_i^{LV-C} := \left\|C_i^I(\cdot;\bar{\beta}_{i+1}^{LV}) - C_i^s(\cdot)\right\|_\infty$ and $\tilde{e}_i^{LP-C} := \left\|C_i^I(\cdot;\bar{\beta}_{i+1}^{LP}) - C_i^s(\cdot)\right\|_\infty$, $\forall i \in \mathcal{I} \setminus \{N-1\}$.

Corollary 8.4. *It holds that* $\tilde{e}_i^{LV-C} \leq \sum_{j=i}^{N-2} \delta^{j-i}(\delta e_{j+1}^*)$ *and* $\tilde{e}_i^{LP-C} \leq \delta^{j-i}\sum_{j=i}^{N-2}\left[\delta(e_{j+1}^* + \bar{e}_{j+1}^{LP})\right]$, $\forall i \in \mathcal{I} \setminus \{N-1\}$.

The error bounds on the errors $\tilde{e}_i^{LV-C}$ and $\tilde{e}_i^{LP-C}$ in Corollary 8.4 are similar to the bounds on the errors $\tilde{e}_i^{LV}$ and $\tilde{e}_i^{LP}$, except that all the bounds of error terms are discounted back by one more stage.

Comparing the error bounds for the CVFA induced by LSMV and 2ALPD with the error bound for the LSMC CVFA is challenging since the regression errors e_{i+1}^* and e_i^{**} originate from different basis functions that approximate the functions V_{i+1}^s and C_i^s. These latter functions are related as $C_i^s(x_{i+1}, F_i^p) \equiv \delta\mathbb{E}^s\left[V_{i+1}^s(x_{i+1}, \tilde{F}_{i+1})|F_i'^p\right]$. We expect C_i^s to be smoother than V_{i+1}^s. It is plausible to expect the inequality that $e_i^{**} \leq \delta e_{i+1}^*$ should hold for similar basis functions and the same number of samples. If, in addition, the inequality $\delta e_{i+1}^* \leq e_i^{**} + \bar{e}_i^{LC}$ is true, then the error bound on $\tilde{e}_i^{LV-C}$ stated in Corollary 8.4 is no worse than the error bound on $\tilde{e}_i^{LC}$ stated in Theorem 8.3. Although it is difficult to characterize when this inequality is true, this resulting ordering of these error bounds provides some support for our numerical observations made in §7 that even the stable LSMC lower bounds do not improve on the stable LSMV lower bounds.

Comparing the error bounds for the LSMC CVFA and the CVFA induced by 2ALPD is more complicated than the analogous comparison for the LSMC CVFA and the CVFA induced by LSMV

because of the presence of $\bar{e}_i^{LP}$ in the 2ALPD error bound. As already discussed, $\bar{e}_i^{LP}$ should increase with P and $\bar{e}_i^{LC}$ should decrease with P. In addition, for sufficiently large P, if the inequalities $\bar{e}_i^{LC} \leq \delta \bar{e}_{i+1}^{LP}$ and $e_i^{**} \leq \delta e_{i+1}^{*}$ hold, then the error bound on $\tilde{e}_i^{LC}$ stated in Theorem 8.3 is no worse than the error bound on $\tilde{e}_i^{LP-C}$ stated in Corollary 8.4. Again, it is difficult to characterize when these inequalities are true. However, this reasoning and this conclusion provide some support for our numerical observations made in §7. Specifically, that the ALPD lower bound estimates outperform the LSMC ones on a few instances for P = 1,000 but the LSMC lower bound estimates outperform the ALPD ones on all the instances with P = 10,000.

9 Surrogate Relaxation Perspective on LSM

To our knowledge there are no known structural connections between the LSM and ALP approaches used in our research. Assuming non negative basis functions, we now prove a structural connection between LSMV and our ALPs: ALPD and ALP. The assumption of non negative basis functions is mild and rather natural in the context of option valuation problems.

The structural relationship between ALPD or ALP and LSMV is based on the concept of surrogate constraints, introduced by Glover (1968). Let $\{g_l(y) \geq z_l,\ l = 1, \ldots, L\}$, be a set of constraints, where $y \in \mathbb{R}$, $g_l : y \mapsto \mathbb{R}$, and $z_l \in \mathbb{R}$. We express as $g(y) \geq z$ this set of constraints in vector form. Let $u \in \mathbb{R}_+^L$ be a vector of non-negative multipliers. A *surrogate constraint* $ug(y) \geq uz$ is a combination of the constraint set $g(y) \geq z$ using the vector of multipliers u. A *surrogate relaxation* of the constraint set $g(y) \geq z$ is a set of surrogate constraints.

Following §8, let $\bar{\beta}_i^{LV}$, $\forall i \in \mathcal{I} \setminus \{0\}$, be the solution obtained by LSMV, and Y be the $(|\mathcal{X}_i| \times P) \times B_i^V$ regression matrix, where $Y_{(x_i,p),b} := \phi_{i,b}(x_i, F_i^p)$. We start by defining a surrogate relaxation of ALPD_i. Consider the following linear program, which we label ALPSR_i (SR stands for surrogate relaxation):

$$\min_{\beta_i} \sum_{x_i \in \mathcal{X}_i, p \in \{1,\ldots,P\}} \left(\sum_{b \in \{1,\ldots,B_i^V\}} \phi_{i,b}(x_i, F_i^p) \right) (\Phi_i \beta_i)(x_i, F_i^p) \tag{26_i}$$

$$\text{s.t.} \sum_{x_i \in \mathcal{X}_i, p \in \{1,\ldots,P\}} \phi_{i,b}(x_i, F_i^p)(\Phi_i \beta_i)(x_i, F_i^p) \geq \sum_{x_i \in \mathcal{X}_i, p \in \{1,\ldots,P\}} \phi_{i,b}(x_i, F_i^p) \left[\mathcal{H}_{i,x_i,F_i^p}^V (\Phi_{i+1} \bar{\beta}_{i+1}^{SR}) \right], \quad \forall b \in \{1, \ldots, B_i^V\}, \tag{27_i}$$

where the objective function is (17_i) specified with $c_i(x_i, F_i{}^p) = \sum_{b \in \{1,\ldots,B_i^V\}} \phi_{i,b}(x_i, F_i^p)$; the constraint set includes B_i^V surrogate constraints of (18_i) with the surrogate constraint corresponding to $b \in \{1, \ldots, B_i^V\}$ derived using the multipliers $\phi_{i,b}(x_i, F_i^p)$, $\forall (x_i, p) \in \mathcal{X}_i \times \{1, \ldots, P\}$; and $\bar{\beta}_{i+1}^{SR}$ is the optimal solution to ALPSR_{i+1}. Theorem 9.1 uses the solutions $\bar{\beta}_i^{SR}$ to the linear programs ALPSR_i, $\forall i \in \mathcal{I} \setminus \{0\}$, to connect LSMV and ALPD. A result analogous to Theorem 9.1 can be proved for LSMC.

Theorem 9.1. *Suppose Y has full column rank. It holds that $\bar{\beta}_i^{SR} = \bar{\beta}_i^{LV}$, $\forall i \in \mathcal{I} \setminus \{0\}$.*

Theorem 9.1 states that the LSMV weight vector $\bar{\beta}_i^{LV}$ is the unique optimal solution to ALPSR_i.

This result implies that performing a specific surrogate relaxation of ALPD is equivalent to removing the constraints from ALPD and at the same time changing the 1,c-norm used in its objective function to a 2-norm, because this yields LSMV. Theorem 8.2 attributes the additional error in the ALPD VFA over the LSMV VFA to the ALPD constraints. Further, our numerical results in §7 show LSMV to be superior to ALPD in terms of the quality of estimated lower and upper bounds. Thus, these considerations *suggest* that the surrogate relaxation used to derive ALPSR_i from ALPD is useful, and motivate further research into surrogate relaxations of ALPD.

We now establish a structural connection between LSMV and ALP in Theorem 9.2. Given $\hat{c} \in \mathbb{R}^{(P \times \sum_{i \in \mathcal{I} \setminus \{0\}} |\mathcal{X}_i|)}$, define the convex program

$$\min_{\beta} \sum_{i \in \{\mathcal{I} \setminus 0\}, x_i \in \mathcal{X}_i, p \in \{1,\ldots,P\}} \hat{c}_i(x_i, F_i^p)(\Phi_i \beta_i)(x_i, F_i^p) \tag{29}$$

$$\text{s.t.} \sum_{x_i \in \mathcal{X}_i, p \in \{1,\ldots,P\}} \phi_{i,b}(x_i, F_i^p)(\Phi_i \beta_i)(x_i, F_i^p) \geq \sum_{x_i \in \mathcal{X}_i, p \in \{1,\ldots,P\}} \phi_{i,b}(x_i, F_i^p) \left[\mathcal{H}^V_{i,x_i,F_i^p}(\Phi_{i+1}\beta_{i+1})\right], \quad \forall (i,b) \in \{\mathcal{I} \setminus \{0\}\} \times \{1,\ldots,B_i^V\}, \tag{31}$$

where the objective function is identical to the one of ALP, and the constraint corresponding to the pair (i,b) in (31) is obtained by applying a surrogate relaxation using the multipliers $\phi_{i,b}(x_i, F_i^p)$, $\forall (x_i, p) \in \mathcal{X}_i \times \{1,\ldots,P\}$, to the equivalent reformulation $\mathcal{H}^V_{i,x_i,F_i^p}(\Phi_{i+1}\beta_{i+1})$ of subsets of the ALP constraints corresponding to stage i. Thus, (29)-(31) is a surrogate relaxation of ALP.

Theorem 9.2. *Suppose that the regression matrix Y has full column rank. Then, there exists a $\hat{c}$ such that $\bar{\beta}_i^{LV}$, $\forall i \in \mathcal{I} \setminus \{0\}$, is the unique optimal solution to* (29)-(31).

Theorem 9.2 shows that LSMV is the unique optimal solution to a surrogate relaxation of ALP for some objective function, not necessarily equal to (26_i). This result motivates further research into surrogate relaxations of approximate linear programs.

10 Conclusions

We develop a value function based LSM method (LSMV) and an ALP method (ALPD) for valuing multiple exercise options when using term structure models. We benchmark our methods against the standard continuation value function based LSM method (LSMC) on realistic energy swing and storage option instances. We find that LSMV outperforms ALPD in terms of both valuation accuracy and computational requirement. The tightest LSMV lower and upper bound estimates are comparable to the ones obtained with LSMC. However, LSMV computes a VFA and estimates its lower bounds faster than LSMC computes a CVFA and estimates its lower bounds. Further, estimating upper bounds with LSMV is two orders of magnitude faster than estimating these bounds with LSMC. We also provide some theoretical support for our numerical results and a new structural connection between the LSMV method and both our ALPD method and ALP. Our research provides strong support for the use of LSMV over LSMC and ALPD for pricing multiple exercise options and also motivates further research into surrogate relaxations of ALPs.

Acknowledgments

This research is based upon work supported by the National Science Foundation under grant No. 1129163.

A Proofs

A.1 Proof for §6

Proof of Proposition 6.1. For any stage $i \in \mathcal{I}$, ALPD$_i$ has a bounded optimal solution value if the right hand sides of the constraints (18_i) are bounded, since they enforce a lower bound on each term $(\Phi_i\beta_i)(x_i, F_i^p)$ in the objective function, and $c_i(x_i, F_i^p) > 0$, $\forall (x_i, F_i^p) \in \mathcal{X}_i \times \{1, \ldots, P\}$. The right hand sides of constraints (18_i) at stage $N-1$ are bounded since they are defined by the reward function evaluated over a finite number of sampled states. Boundedness of the right hand sides of constraints (18_i) at every preceding stage follows from a straightforward induction.

To prove basic feasibility of the ALPD solution, notice that the unrestricted variables β_i, $\forall i \in \mathcal{I} \setminus \{0\}$ will be basic in any basic feasible solution (see Definition 2.9 in Bertsimas and Tsitsiklis 1997) to (14)-(16) since they have no bounds. In particular, a basic feasible solution of (14)-(16) is characterized by $\sum_{i\in\mathcal{I}} B_i^V$ non basic slack variables. The optimal solution $\bar{\beta}_i$ of (17_i)-(18_i) at stage i has B_i^V non basic slack variables. Thus, $\bar{\beta}_i$, $\forall i \in \mathcal{I} \setminus \{0\}$ is a basic feasible solution to (14)-(16). □

A.2 Proofs for §8

Proof of Proposition 8.1. For ease of notation define $\eta = (x_i, F_p^i)$, $c(\eta) = c_i(\eta)$, $a(\eta) = (\Phi_i\beta_i)(x_i, F_i^p)$ and $b(\eta) = \mathcal{H}^V_{i,x_i,F_i^p}(\Phi_{i+1}\bar{\beta}_{i+1})$. It holds that

$$\min_{\beta_i}\left\{\sum_{\eta} c(\eta)\cdot|a(\eta)-b(\eta)| : (23_i)\right\} = \min_{\beta_i}\left\{\sum_{\eta} c(\eta)\cdot(a(\eta)-b(\eta)) : (23_i)\right\} \tag{32}$$

$$= \min_{\beta_i}\left\{\sum_{\eta} c(\eta)\cdot a(\eta) : (23_i)\right\} - \sum_{\eta} c(\eta)\cdot b(\eta). \tag{33}$$

□

The left hand side of equation (32) is the math program (22_i)-(23_i). The right hand side of (32) is obtained by removing the absolute value around the term $a(\eta) - b(\eta)$ in the objective function. This is possible since $a(\eta) \geq b(\eta)$ is enforced by the constraints (23_i). Since $b(\eta)$ is a known constant we can bring it out of the minimization to obtain (33). The first term in (33) is ALPD$_i$. This shows that solving ALPD$_i$ is equivalent to the math program (22_i)-(23_i) in the sense that an optimal solution to one of these math programs is also optimal to the other.

Proof of Theorem 8.2. We start with LSMV. At stage $N-1$ we have

$$\tilde{e}^{LV}_{N-1} = \left\|\hat{V}_{N-1}(\cdot;\bar{\beta}^{LV}_{N-1}) - V^s_{N-1}(\cdot)\right\|_\infty \leq \left\|\hat{V}_{N-1}(\cdot;\bar{\beta}^{LV}_{N-1}) - V^s_{N-1}(\cdot)\right\|_2 = \left\|\Pi_2 V^s_{N-1}(\cdot) - V^s_{N-1}(\cdot)\right\|_2 = e^*_{N-1}.$$

We obtain the first inequality from the well known relation $\|g_i\|_2 \geq \|g_i\|_\infty$. The second equality is obtained by noticing that LSMV performs a regression against V^s_{N-1} to compute $\bar{\beta}^{LV}_{N-1}$ at stage $N-1$ (see Algorithm 1). For stage $i \in \mathcal{I} \setminus \{0, N-1\}$, we have

$$\tilde{e}^{LV}_i \quad = \quad \left\|\hat{V}_i(\cdot;\bar{\beta}^{LV}_i) - V^s_i(\cdot)\right\|_\infty$$

$$
\begin{aligned}
&\leq \left\|\hat{V}_i(\cdot;\bar{\beta}_i^{LV}) - \Pi_2 V_i^s(\cdot)\right\|_\infty + \|\Pi_2 V_i^s(\cdot) - V_i^s\|_\infty \qquad (34)\\
&\leq \left\|\hat{V}_i(\cdot;\bar{\beta}_i^{LV}) - \Pi_2 V_i^s(\cdot)\right\|_\infty + \|\Pi_2 V_i^s(\cdot) - V_i^s\|_2\\
&= \left\|\hat{V}_i(\cdot;\bar{\beta}_i^{LV}) - \Pi_2 V_i^s(\cdot)\right\|_\infty + e_i^*.
\end{aligned}
$$

Inequality (34) is true by the triangle inequality property of norms. Now we bound $\hat{e}_i := \left\|\hat{V}_i(\cdot;\bar{\beta}_i^{LV}) - \Pi_2 V_i^s(\cdot)\right\|_\infty$. For functions W_1 and W_2 with the same domain, it holds that $\|\Pi_2 W_1 - \Pi_2 W_2\|_\infty = \|\Pi_2(W_1 - W_2)\|_\infty \leq \|W_1 - W_2\|_\infty$. The first equality holds since the optimal solution to a 2-norm regression problem is linear in its argument. The second inequality follows from a well known property of projection operators. For ease of notation we define $\eta = (x_i - a, \tilde{F}_{i+1})$ and $\bar{\eta} = (x_i - a, \tilde{F}_{i+1}; \bar{\beta}_{i+1}^{LV})$. Then, setting $W_1 = \max_{a\in\mathcal{A}_i(x_i)} r_i(a, F_i^p) + \delta\mathbb{E}^s\left[\hat{V}_{i+1}(\bar{\eta})|F_i'^p\right]$ and $W_2 = \max_{a\in\mathcal{A}_i(x_i)} r_i(a, F_i^p) + \delta\mathbb{E}^s\left[V_{i+1}^s(\eta)|F_i'^p\right]$, we have

$$
\begin{aligned}
\hat{e}_i &\leq \left\|\left(\max_{a\in\mathcal{A}_i(x_i)} r_i(a, F_i^p) + \delta\mathbb{E}^s\left[\hat{V}_{i+1}(\bar{\eta})|F_i'^p\right]\right) - \left(\max_{a\in\mathcal{A}_i(x_i)} r_i(a, F_i^p) + \delta\mathbb{E}^s\left[V_{i+1}^s(\eta)|F_i'^p\right]\right)\right\|_\infty\\
&\leq \delta\left\|\max_{a\in\mathcal{A}_i(x_i)} \mathbb{E}^s\left[\hat{V}_{i+1}(\bar{\eta}) - V_{i+1}^s(\eta)|F_i'^p\right]\right\|_\infty \qquad (35)\\
&= \delta \max_{x_i\in\mathcal{X}_i, p\in\{1,\ldots,P\}}\left|\max_{a\in\mathcal{A}_i(x_i)} \mathbb{E}^s\left[\left(\hat{V}_{i+1}(\bar{\eta}) - V_{i+1}^s(\eta)\right)|F_i'^p\right]\right|\\
&\leq \delta \max_{x_i\in\mathcal{X}_i}\left|\max_{a\in\mathcal{A}_i(x_i), p\in\{1,\ldots,P\}}\left(\hat{V}_{i+1}(\bar{\eta}) - V_{i+1}^s(\eta)\right)\right|\\
&\leq \delta \max_{x_i\in\mathcal{X}_i}\max_{a\in\mathcal{A}_i(x_i), p\in\{1,\ldots,P\}}\left|\hat{V}_{i+1}(\bar{\eta}) - V_{i+1}^s(\eta)\right| \qquad (36)\\
&\leq \delta \max_{x_{i+1}\in\mathcal{X}_{i+1}, p\in\{1,\ldots,P\}}\left|\hat{V}_{i+1}(x_{i+1}, F_{i+1}^p; \bar{\beta}_{i+1}^{LV}) - V_{i+1}^s(x_{i+1}, F_{i+1}^p)\right| = \delta\tilde{e}_{i+1}^{LV}. \qquad (37)
\end{aligned}
$$

Inequality (35) is obtained by noting that $\max_{s\in S} g(s) - \max_{s\in S} z(s) \leq \max_{s\in S}(g(s) - z(s))$. We obtain inequality (36) using the relation $|\max_{s\in S} g(s)| \leq \max_{s\in S}|g(s)|$. The rest follows from the definition of $\|\cdot\|_\infty$. Thus, we have $\tilde{e}_i^{LV} \leq \delta\tilde{e}_{i+1}^{LV} + e_i^*$.

Next we consider 2ALPD. At stage $N-1$, we have

$$
\begin{aligned}
\tilde{e}_{N-1}^{LP} &= \left\|\hat{V}_{N-1}(\cdot;\bar{\beta}_{N-1}^{LP}) - V_{N-1}^s(\cdot)\right\|_\infty\\
&\leq \left\|\hat{V}_{N-1}(\cdot;\bar{\beta}_{N-1}^{LP}) - \Pi_2\mathcal{H}_{N-1,\cdot,\cdot}^{V,s}\left(\hat{V}_N(\cdot;\bar{\beta}_N^{LP})\right)\right\|_\infty + \left\|\Pi_2\mathcal{H}_{N-1,\cdot,\cdot}^{V,s}\left(\hat{V}_N(\cdot;\bar{\beta}_N^{LP})\right) - V_{N-1}^s(\cdot)\right\|_\infty\\
&\leq \left\|\hat{V}_{N-1}(\cdot;\bar{\beta}_{N-1}^{LP}) - \Pi_2\mathcal{H}_{N-1,\cdot,\cdot}^{V,s}\left(\hat{V}_N(\cdot;\bar{\beta}_N^{LP})\right)\right\|_\infty + \left\|\Pi_2 V_{N-1}^s(\cdot) - V_{N-1}^s(\cdot)\right\|_2\\
&= \bar{e}_{N-1}^{LP} + e_{N-1}^*.
\end{aligned}
$$

At stage i, we have

$$
\begin{aligned}
\tilde{e}_i^{LP} &= \left\|\hat{V}_i(\cdot;\bar{\beta}_i^{LP}) - V_i^s(\cdot)\right\|_\infty\\
&\leq \left\|\hat{V}_i(\cdot;\bar{\beta}_i^{LP}) - \Pi_2\mathcal{H}_{i,\cdot,\cdot}^{V,s}\left(\hat{V}_{i+1}(\cdot;\bar{\beta}_{i+1}^{LP})\right)\right\|_\infty + \left\|\Pi_2\mathcal{H}_{i,\cdot,\cdot}^{V,s}\left(\hat{V}_{i+1}(\cdot;\bar{\beta}_{i+1}^{LP})\right) - V_i^s(\cdot)\right\|_\infty\\
&\leq \bar{e}_i^{LP} + \left\|\Pi_2\mathcal{H}_{i,\cdot,\cdot}^{V,s}\left(\hat{V}_{i+1}(\cdot;\bar{\beta}_{i+1}^{LP})\right) - \Pi_2 V_i^s(\cdot)\right\|_\infty + \|\Pi_2 V_i^s(\cdot) - V_i^s(\cdot)\|_\infty\\
&\leq \bar{e}_i^{LP} + \delta\tilde{e}_i^{LP} + e_i^*. \qquad (38)
\end{aligned}
$$

We obtain inequality (38) using the definitions of $\bar{e}_i^{LP}$ and e_i^*, and the inequality $\left\|\hat{V}_i(\cdot;\beta_i') - \Pi_2 V_i^s(\cdot)\right\|_\infty \leq$

$\delta\tilde{e}_i^{LP}$, which can be derived following steps analogous to (35)-(37). The remaining statements of the theorem follow immediately. □

Proof of Theorem 8.3. At stage $N-2$, we have

$$\begin{aligned}
\tilde{e}_{N-2}^{LC} &= \left\|\hat{C}_{N-2}(\cdot;\bar{\theta}_{N-2}) - C_{N-2}^s(\cdot)\right\|_\infty \\
&\leq \left\|\hat{C}_{N-2}(\cdot;\bar{\theta}_{N-2}) - \Pi_2\mathcal{H}_{N-2,\cdot,\cdot}^{C,s}\left(\hat{C}_{N-1}(\cdot;\bar{\theta}_{N-1}^{LC})\right)\right\|_\infty + \left\|\Pi_2\mathcal{H}_{N-2,\cdot,\cdot}^{C,s}\left(\hat{C}_{N-1}(\cdot;\bar{\theta}_{N-1}^{LC})\right) - C_{N-2}^s(\cdot)\right\|_\infty \\
&\leq \bar{e}_{N-2}^{LC} + \left\|\Pi_2 C_{N-2}^s(\cdot) - C_{N-2}^s(\cdot)\right\|_2 = \bar{e}_{N-2}^{LC} + e_{N-2}^{**}.
\end{aligned}$$

At stage $i \in \mathcal{I}\setminus\{N-1, N-2\}$, it holds that

$$\begin{aligned}
\tilde{e}_i^{LC} &= \left\|\hat{C}_i(\cdot;\bar{\theta}_i) - C_i^s(\cdot)\right\|_\infty \\
&\leq \left\|\hat{C}_i(\cdot;\bar{\theta}_i) - \Pi_2 C_i^s(\cdot)\right\|_\infty + \left\|\Pi_2 C_i^s(\cdot) - C_i^s(\cdot)\right\|_\infty \\
&\leq \left\|\hat{C}_i(\cdot;\bar{\theta}_i) - \Pi_2 C_i^s(\cdot)\right\|_\infty + e_i^{**} \\
&\leq \left\|\hat{C}_i(\cdot;\bar{\theta}_i) - \Pi_2\mathcal{H}_{i,\cdot,\cdot}^{C,s}\left(\hat{C}_{i+1}(\cdot;\bar{\theta}_{i+1}^{LC})\right)\right\|_\infty + \left\|\Pi_2\mathcal{H}_{i,\cdot,\cdot}^{C,s}\left(\hat{C}_{i+1}(\cdot;\bar{\theta}_{i+1}^{LC})\right) - \Pi_2 C_i^s(\cdot)\right\|_\infty + e_i^{**} \\
&= \bar{e}_i^{LC} + \left\|\Pi_2\mathcal{H}_{i,\cdot,\cdot}^{C,s}\left(\hat{C}_{i+1}(\cdot;\bar{\theta}_{i+1}^{LC})\right) - \Pi_2 C_i^s(\cdot)\right\|_\infty + e_i^{**}.
\end{aligned}$$

Further, we can show that $\left\|\Pi_2\mathcal{H}_{i,\cdot,\cdot}^{C,s}\left(\hat{C}_{i+1}(\cdot;\bar{\theta}_{i+1}^{LC})\right) - \Pi_2 C_i^s(\cdot)\right\|_\infty \leq \delta\tilde{e}_{i+1}^{LC}$ following steps similar to those required to bound the term $\left\|\hat{V}_i(\cdot;\bar{\beta}_i^{LV}) - \Pi_2 V_i^s(\cdot)\right\|_\infty$ in Theorem 8.2. The rest of the statements in the theorem follow immediately. □

Proof of Corollary 8.4. We have

$$\begin{aligned}
\tilde{e}_i^{LV-C} &= \left\|C_i^I(\cdot;\bar{\beta}_{i+1}^{LV}) - C_i^s(\cdot)\right\|_\infty = \delta\left\|\mathbb{E}\left[\hat{V}_i(x_{i+1},\tilde{F}_{i+1};\bar{\beta}_{i+1}^{LV}) - V_i^s(x_{i+1},\tilde{F}_{i+1})|F_i'^p\right]\right\|_\infty \\
&\leq \delta\left\|\hat{V}_i(\cdot;\bar{\beta}_{i+1}^{LV}) - V_{i+1}^s(\cdot)\right\|_\infty = \delta\tilde{e}_{i+1}^{LV}.
\end{aligned}$$

The bound in the corollary for the error term $\tilde{e}_i^{LV-C}$ can be obtained by using the bound on the error term $\tilde{e}_{i+1}^{LV}$ from Theorem 8.2 and the inequality $\tilde{e}_i^{LV-C} \leq \delta\tilde{e}_{i+1}^{LV}$ just established. The bound on the error term $\tilde{e}_i^{LP-C}$ can be established in an identical manner. □

A.3 Proofs for §9

Proof of Theorem 9.1. At stage i the ALPD constraint set (18_i) is $(\Phi_i\beta_i)(x_i,F_i^p) \geq \mathcal{H}_{i,x_i,F_i^p}^V(\Phi_{i+1}\bar{\beta}_{i+1}),\ \ \forall x_i \in \mathcal{X}_i, p \in \{1,\ldots,P\}$. For a given $b \in \{1,\ldots,B_i^V\}$, consider the surrogate multipliers $\phi_{i,b}(x_i,F_i^p) \geq 0, \forall(x_i,p) \in \mathcal{X}_i \times \{1,\ldots,P\}$. Taking a positive linear combination of (18_i) using these surrogate multipliers results in the surrogate constraint

$$\sum_{x_i\in\mathcal{X}_i, p\in\{1,\ldots,P\}} \phi_{i,b}(x_i,F_i^p)(\Phi_i\beta_i)(x_i,F_i^p) \geq \sum_{x_i\in\mathcal{X}_i, p\in\{1,\ldots,P\}} \phi_{i,b}(x_i,F_i^p)\left[\mathcal{H}_{i,x_i,F_i^p}^V(\Phi_{i+1}\bar{\beta}_{i+1})\right].$$

Repeating this for each $b \in \{1,\ldots,B_i^V\}$ gives us the constraint set (27_i).

The linear program (26_i)-(27_i) is thus a surrogate relaxation of (17_i)-(18_i) with B_i^V unrestricted variables, $\beta_{i,b}, \forall b \in \{1,\ldots,B_i^V\}$, and the same number of constraints. Since the variables β_i are unrestricted they can

be treated as basic in any basic feasible solution (see Definition 2.9 in Bertsimas and Tsitsiklis 1997), and a basic feasible solution has B_i^V non basic slack variables. Thus, the inequality constraint set (27_i) holds as an equality at a basic feasible solution.

To connect (26_i)-(27_i) to the 2-norm regression problem solved by LSMV we define the residual $\epsilon_{i,x_i,p}(\beta_i) := (\Phi_i\beta_i)(x_i, F_i^p) - \mathcal{H}^V_{i,x_i,F_i^p}(\Phi_{i+1}\bar{\beta}_{i+1})$. It is well known that the 2-norm regression solution $\beta_i' := \text{argmin}_{\beta_i} \sum_{(x_i,p)} [\epsilon_{i,x_i,p}(\beta_i)]^2$ is the unique solution to the inequalities (27_i), expressed as equalities. The uniqueness is a consequence of the full rank assumption on the regression matrix Y.

Now we show that $\bar{\beta}_i^{LV}$, $\forall i \in \mathcal{I} \setminus \{0\}$, is the unique optimal solution to the linear programs (26_i)-(27_i) $\forall i \in \mathcal{I}$ that are solved in a backward recursive manner. We argue this in two steps: First, when $c_i(x_i, F_i^p) = \sum_{b\in\{1,\ldots,B_i^V\}} \phi_{i,b}(x_i, F_i^p)$, $\forall x_i \in \mathcal{X}_i, p \in \{1,\ldots,P\}$, the unique basic feasible solution characterized above is optimal, since the objective function of (26_i)-(27_i) minimizes the sum of the left hand side terms of the constraints (27_i). Second, applying an induction argument starting from stage $N-1$ and moving backward to stage 1 shows that this unique optimal solution is indeed $\bar{\beta}_i^{LV}$, $\forall i \in \mathcal{I} \setminus \{0\}$. □

Proof of Theorem 9.2. At stage $i \in \mathcal{I} \setminus \{0\}$, the ALP constraint can be equivalently reformulated as $(\Phi_i\beta_i)(x_i, F_i^p) \geq \mathcal{H}^V_{i,x_i,F_i^p}(\Phi_{i+1}\beta_{i+1}), \forall i \in \mathcal{I}, x_i \in \mathcal{X}_i, p \in \{1,\ldots,P\}$. These inequalities are convex constraints since the right hand side is a maximum over affine functions. Then, for a given $b \in \{1,\ldots,B_i^V\}$, consider the surrogate multipliers $\phi_{i,b}(x_i, F_i^p) \geq 0$, $\forall (x_i, p) \in \mathcal{X}_i \times \{1,\ldots,P\}$. Taking a positive linear combination of the stage i ALP constraints with these multipliers results in the convex surrogate constraint

$$\sum_{x_i\in\mathcal{X}_i, p\in\{1,\ldots,P\}} \phi_{i,b}(x_i, F_i^p)(\Phi_i\beta_i)(x_i, F_i^p) \geq \sum_{x_i\in\mathcal{X}_i, p\in\{1,\ldots,P\}} \phi_{i,b}(x_i, F_i^p) \left[\mathcal{H}^V_{i,x_i,F_i^p}(\Phi_{i+1}\beta_{i+1})\right].$$

Repeating this for each $b \in \{1,\ldots,B_i^V\}$ and then for each stage $i \in \mathcal{I} \setminus \{0\}$ gives us the constraints (31). Due to the full rank assumption on Y it is easy to verify through an induction argument starting from stage $N-1$ that $\bar{\beta}_i^{LV}$, $\forall i \in \mathcal{I} \setminus \{0\}$ is the unique solution to the equality system defined by the inequalities (31) holding as equalities.

Moreover, $\bar{\beta}_i^{LV}$, $\forall i \in \mathcal{I} \setminus \{0\}$ is a boundary point of the convex feasible region (31). Then it follows from the linear support function characterization of convex sets (see §2.4 in Hiriart-Urruty and Lemaréchal 2001) that there exists a $\hat{c}$ such that $\bar{\beta}_i^{LV}$, $\forall i \in \mathcal{I} \setminus \{0\}$ is an optimal solution to the convex program (29)-(31). □

References

Adelman, D. 2007. Dynamic bid prices in revenue management. *Operations Research* **55**(4) 647–661.

Andersen, L., M. Broadie. 2004. Primal-dual simulation algorithm for pricing multidimensional American options. *Management Science* **50**(9) 1222–1234.

Barbieri, A., M.B. Garman. 1996. Putting a price on swings. *Energy and Power Risk Management* **1**(6) 17–19.

Bellman, R., S. Dreyfus. 1959. Functional approximations and dynamic programming. *Mathematical Tables and Other Aids to Computation* **13**(68) 247–251.

Bender, C. 2011. Dual pricing of multi-exercise options under volume constraints. *Finance and Stochastics* **15**(1) 1–26.

Bertsekas, P. B. 2007. *Dynamic Programming and Optimal Control*, vol. 2. 3rd ed. Athena Scientific, Nashua, New Hampshire, USA.

Bertsimas, D., J.N. Tsitsiklis. 1997. *Introduction to Linear Optimization*. Athena Scientific, MA, USA.

Bjerksund, P., G. Stensland. 2011. Closed form spread option valuation. *Quantitative Finance*, Forthcoming.

Boogert, A., C. De Jong. 2008. Gas storage valuation using a Monte Carlo method. *The Journal of Derivatives* **15**(3) 81–98.

Boogert, A., D. Mazières. 2011. A radial basis function approach to gas storage valuation. Working paper, Birkbeck University of London.

Brown, D. B., J. E. Smith, P. Sun. 2010. Information relaxations and duality in stochastic dynamic programs. *Operations Research* **58**(4) 1–17.

Carriere, J.F. 1996. Valuation of the early-exercise price for options using simulations and nonparametric regression. *Insurance: Mathematics and Economics* **19**(1) 19–30.

Chandramouli, S.S., M. Haugh. 2012. A unified approach to multiple stopping and duality. *Operations Research Letters* **Forthcoming**.

Clewlow, L., C. Strickland. 2000. *Energy Derivatives: Pricing and Risk Management*. Lacima, London, UK.

Cortazar, G., M. Gravet, J. Urzua. 2008. The valuation of multidimensional American real options using the LSM simulation method. *Computers & Operations Research* **35**(1) 113–129.

Cortazar, G., E.S. Schwartz. 1994. The valuation of commodity contingent claims. *The Journal of Derivatives* **1**(4) 27–39.

CPLEX, I. 2010. 12.2. User's Manual.

de Farias, D. P., B. Van Roy. 2003. The linear programming approach to approximate dynamic programming. *Operations Research* **51**(6) 850–865.

Detemple, J. 2006. *American-style Derivatives: Valuation and Computation*. Taylor & Francis, London, UK.

Devalkar, S.K., R. Anupindi, A. Sinha. 2011. Integrated optimization of procurement, processing, and trade of commodities. *Operations Research* **59**(6) 1369–1381.

Eydeland, A., K. Wolyniec. 2003. *Energy and Power Risk Management*. John Wiley and Sons Inc., Hoboken, NJ, USA.

Farias, V. F., B. Van Roy. 2007. An approximate dynamic programming approach to network revenue management. Working paper, Stanford Univ.

Glasserman, P. 2004. *Monte Carlo Methods in Financial Engineering*. Springer, New York, NY, USA.

Glasserman, P., B. Yu. 2004. Simulation for American options: Regression now or regression later. H. Niederreiter, ed., *Monte Carlo and Quasi-Monte Carlo Methods 2002*. Springer-Verlag, Berlin, Germany, 213–226.

Glover, F. 1968. Surrogate constraints. *Operations Research* **16**(4) 741–749.

Gyurko, L.G., B. Hambly, J. H. Witte. 2011. Monte Carlo methods via a dual approach for some discrete time stochastic control problems. *ArXiv e-prints*.

Hiriart-Urruty, J.B., C. Lemaréchal. 2001. *Fundamentals of Convex Analysis*. Springer-Verlag, Berlin, Germany.

Ho, T.S.Y., S.B. Lee. 1986. Term structure movements and pricing interest rate contingent claims. *Journal of Finance* **41**(5) 1011–1029.

Jaillet, P., E.I. Ronn, S. Tompaidis. 2004. Valuation of commodity-based swing options. *Management Science* **50**(7) 909–921.

Lai, G., F. Margot, N. Secomandi. 2010. An approximate dynamic programming approach to benchmark practice-based heuristics for natural gas storage valuation. *Operations Research* **58**(3) 564–582.

Longstaff, F. A., E. S. Schwartz. 2001. Valuing American options by simulation: A simple least-squares approach. *Review of Financial Studies* **14**(1) 113–147.

Maragos, S. 2002. Valuation of the operational flexibility of natural gas storage reservoirs. E. Ronn, ed., *Real Options and Energy Management Using Options Methodology to Enhance Capital Budgeting Decisions*. Risk Publications, London, UK, 431–456.

Meinshausen, N., B. M. Hambly. 2004. Monte carlo methods for the valuation of multiple-exercise options. *Mathematical Finance* **14**(4) 557–583.

Moallemi, C. C., S. Kumar, B. Van Roy. 2008. Approximate and data-driven dynamic programming for queuing networks. Working paper, Stanford Univ.

Nadarajah, S., F. Margot, N. Secomandi. 2012. Approximate dynamic programs for natural gas storage valuation based on approximate linear programming relaxations. Working paper, Carnegie Mellon Univ.

Powell, W. B. 2011. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. 2nd ed. John Wiley & Sons, Hoboken, New Jersey, USA.

Rogers, L. C. G. 2002. Monte Carlo valuation of American options. *Mathematical Finance* **12**(3) 271–286.

Schwartz, E., J.E. Smith. 2000. Short-term variations and long-term dynamics in commodity prices. *Management Science* **46**(7) 893–911.

Schwartz, E.S. 1997. The stochastic behavior of commodity prices: Implications for valuation and hedging. *Journal of Finance* **52**(3) 923–974.

Schweitzer, P. J., A. Seidmann. 1985. Generalized polynomial approximations in Markovian decision processes. *Journal of Mathematical Analysis and Applications* **110**(2) 568–582.

Secomandi, N. 2010. Optimal commodity trading with a capacitated storage asset. *Management Science* **56**(3) 1090–1049.

Secomandi, N. 2012. The role of price spreads and reoptimization in the real option management of commodity storage assets. Working paper, Carnegie Mellon Univ.

Secomandi, N., G. Lai, F. Margot, A. Scheller-Wolf, D. Seppi. 2012. Merchant commodity storage and term structure model error. Working paper, Carnegie Mellon Univ.

Thompson, M. 2012. Natural gas storage valuation, optimization, market and credit risk management. Working paper, Queens Univ.

Tong, C., H. Topaloglu. 2011. On approximate linear programming approach for network revenue management problems. Working paper, Cornell Univ.

Tsitsiklis, J.N., B. Van Roy. 2001. Regression methods for pricing complex American-style options. *IEEE Transactions on Neural Networks* **12**(4) 694–703.

Veronesi, P. 2010. *Fixed Income Securities: Valuation, Risk, and Risk Management*. John Wiley and Sons Inc., Hoboken, NJ, USA.

Vossen, T., D. Zhang. 2012. A dynamic disaggregation approach to approximate linear programs for network revenue management. Working paper, Univ. of Colorado at Boulder.

Wu, O.Q., D.D. Wang, Z. Qin. 2012. Seasonal energy storage operations with limited flexibility: The price-adjusted rolling intrinsic policy. *Manufacturing & Service Operations Management* **14**(3) 455–471.